

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

LA COMMUNICATION SANS FIL DANS UN RÉSEAU ÉLECTRIQUE
INTELLIGENT (SMART GRID) – MÉTHODOLOGIE DE DÉVELOPPEMENT

MÉMOIRE

PRÉSENTÉ

COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN INFORMATIQUE

PAR

BENAOUDA BAGHDAD

AVRIL 2013

UNIVERSITÉ DU QUÉBEC À MONTRÉAL
Service des bibliothèques

Avertissement

La diffusion de ce mémoire se fait dans le respect des droits de son auteur, qui a signé le formulaire *Autorisation de reproduire et de diffuser un travail de recherche de cycles supérieurs* (SDU-522 – Rév.01-2006). Cette autorisation stipule que «conformément à l'article 11 du Règlement no 8 des études de cycles supérieurs, [l'auteur] concède à l'Université du Québec à Montréal une licence non exclusive d'utilisation et de publication de la totalité ou d'une partie importante de [son] travail de recherche pour des fins pédagogiques et non commerciales. Plus précisément, [l'auteur] autorise l'Université du Québec à Montréal à reproduire, diffuser, prêter, distribuer ou vendre des copies de [son] travail de recherche à des fins non commerciales sur quelque support que ce soit, y compris l'Internet. Cette licence et cette autorisation n'entraînent pas une renonciation de [la] part [de l'auteur] à [ses] droits moraux ni à [ses] droits de propriété intellectuelle. Sauf entente contraire, [l'auteur] conserve la liberté de diffuser et de commercialiser ou non ce travail dont [il] possède un exemplaire.»

A mes parents

À ma femme

À mes filles

À tous mes amis

REMERCIEMENTS

Je voudrais exprimer mes remerciements les plus sincères à toutes les personnes qui m'ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire, ainsi qu'à la réussite de mon cursus universitaire.

En premier lieu, je voudrais adresser mes remerciements à mon directeur de recherche, Professeur Abdellatif Obaid qui a initié et dirigé ce travail. Ses suggestions, ses conseils et ses commentaires précieux m'ont permis de surmonter les difficultés et de progresser dans l'accomplissement de mes objectifs de recherche. Il a toute ma reconnaissance pour l'inspiration, l'aide et le temps qu'il a su me consacrer. Sans lui, ce mémoire n'aurait jamais vu le jour.

Mes remerciements s'adressent également à mon codirecteur Professeur Wessam Ajib pour sa générosité et son suivi rigoureux.

J'exprime ma gratitude à tous les professeurs de l'UQAM qui m'ont accompagné pendant ces années et m'ont permis d'avoir une formation de qualité. Je remercie notamment le directeur de programme de Maîtrise en informatique, Professeur Étienne Gagnon.

Je tiens aussi à remercier tout l'ensemble du personnel de l'UQAM, surtout Madame Lise Arsenault et Madame Louise Tremblay pour leur soutien tout au long de la maîtrise.

J'adresse mes plus sincères remerciements à mes parents pour tous les sacrifices consentis afin de me donner les moyens d'avoir cette formation et leur soutien tout au long de ces dures années d'études.

J'adresse également ma profonde reconnaissance accompagnée de mon affection à ma femme Nassira et mes chères filles Ibtihal, Youssra, Rajaa, et Souha.

Enfin, je remercie tous mes amis Elmahdi Driouch, Mohaned Chraïti, Zakaria El Alaoui, Mohamed Slim et Mourad Dahmane, qui m'ont toujours soutenu et encouragé au cours de la réalisation de ce travail.

TABLE DES MATIÈRES

| | |
|---|-----|
| LISTE DES TABLEAUX ET DES FIGURES..... | vii |
| RÉSUMÉ..... | xi |
| INTRODUCTION..... | 1 |
| 0.1 – Introduction générale..... | 1 |
| 0.2– Problématiques | 4 |
| 0.3 – Le rôle des compteurs intelligents..... | 4 |
| 0.4– L’intégration des énergies alternatives..... | 7 |
| 0.5– Objectifs | 7 |
| 0.6– Méthodologie | 8 |
| 0.7 – Organisation du mémoire..... | 9 |
| CHAPITRE I | |
| RÉSEAUX ÉLECTRIQUES INTELLIGENTS | 11 |
| 1.1 – Définitions..... | 11 |
| 1.2 – Gestion et distribution électrique dans un foyer..... | 12 |
| 1.3 – Gestion de l’offre et la demande | 14 |
| 1.4 – Accès et contrôle à distance | 16 |
| 1.5 – Plateformes industrielles | 19 |
| CHAPITRE II | |
| ARCHITECTURES DE COMMUNICATION..... | 23 |
| 2.1 – Introduction | 23 |
| 2.2 – Architecture de déploiement | 24 |
| 2.3 – Les outils de déploiement..... | 27 |
| 2.3.1 – L’environnement du routeur..... | 27 |
| 2.3.2 – Le compteur intelligent | 30 |
| CHAPITRE III | |
| SIMULATION DU CONTRÔLE DE LA RÉPARTITION DE CHARGE | 33 |
| 3.1 – Environnement et paramètres de simulation | 33 |
| 3.2 – Simulation de la consommation | 34 |
| 3.3 – Fichiers des données de consommation | 42 |

| | |
|---|----|
| 3.4 – Coordination en cas de chute de tension | 44 |
| 3.4.1 – La consommation en état OFF | 45 |
| 3.4.2 – Consommation en cas de chute de tension | 47 |
| 3.5 – Contrôle de consommation par foyer et par zone..... | 49 |
| 3.6 – Contrôle à distance par appareil mobile..... | 53 |
| CHAPITRE IV | |
| MÉCANISME DE NÉGOCIATION DE..... | 57 |
| L’OFFRE ET DE LA DEMANDE D’ÉNERGIE..... | 57 |
| 4.1 – Introduction | 57 |
| 4.2 – Travaux connexes..... | 59 |
| 4.3 – Le middleware..... | 64 |
| 4.4 – Structure d’un agent mobile | 66 |
| 4.5 – Le protocole de négociation | 71 |
| 4.6 – Implémentation | 73 |
| CHAPITRE V | |
| ÉVALUATION DU PROTOCOLE D’ÉCHANGE DES AGENTS MOBILES..... | 77 |
| 5.1 – Introduction | 77 |
| 5.2 – Résultats de simulations | 79 |
| 5.2.1 – Au niveau du Routeur 1 | 79 |
| 5.2.2 – Au niveau des compteurs intelligents..... | 81 |
| CONCLUSION..... | 83 |
| ANNEXE | |
| BIBLIOGRAPHIE | 99 |

LISTE DES TABLEAUX ET DES FIGURES

| Tableau | | Page |
|------------|---|------|
| 3.1 | Données de consommation | 42 |
| Figure | | |
| 0.1 | Schéma de production, du transport et de la distribution d'énergie..... | 1 |
| 1.1 | Composantes d'un système de distribution intelligent..... | 11 |
| 1.2 | Infrastructure d'un réseau électrique intelligent | 12 |
| 1.3 | Couches du système IDAPS [25]..... | 13 |
| 1.4 | Flux d'information dans ADAPS [25] | 14 |
| 1.5 | Le cadre multi-agents dans une cellule IDAPS [25]..... | 15 |
| 1.6 | Schéma architectural du système [27] | 16 |
| 1.7 | Le flux de communication dans le service de détection [28]..... | 18 |
| 1.8 | La solution du réseau proposée par CISCO [30] | 20 |
| 2.1 | Le modèle global d'un réseau électrique intelligent | 24 |
| 2.2 | Types de communications dans le réseau | 25 |
| 2.3 | La communication entre régions..... | 26 |
| 2.4 | La communication entre zones et foyers..... | 26 |
| 2.5 | Communication entre un compteur et les appareils électroménagers | 27 |
| 2.6 | Architecture d'un routeur..... | 28 |
| 2.7 | Création de la table routeurs <i>Voisins</i> | 29 |
| 2.8 | Table de foyers connectés au routeur..... | 30 |
| 2.9 | Table des foyers qui fournissent l'énergie dans une zone..... | 30 |
| 2.10 | Communication entre un compteur et les appareils électroménagers_2..... | 31 |

| | | |
|------|---|----|
| 2.11 | Schéma d'un compteur intelligent | 31 |
| 3.1 | Schéma conceptuel d'un système de surveillance de la charge (NIALM) [45]..... | 35 |
| 3.2 | Schéma global de solution proposé par Kuruppu [46]..... | 35 |
| 3.3 | Détection de comportement d'un appareil électroménager [46] | 36 |
| 3.4 | Les périodes et les sous-périodes dans une journée | 37 |
| 3.5 | Le fichier configuration d'un appareil intelligent | 38 |
| 3.6 | Priorité des appareils dans un foyer [43] | 39 |
| 3.7 | Consommation énergétique des appareils dans une région..... | 41 |
| 3.8 | Communication entre un compteur intelligent et les appareils électroménagers..... | 45 |
| 3.9 | La consommation des quatre appareils ; le cas d'un réseau classique | 46 |
| 3.10 | La consommation des quatre appareils ; le cas d'un réseau intelligent | 46 |
| 3.11 | La consommation des appareils ; le cas d'un réseau intelligent | 48 |
| 3.12 | La consommation des appareils ; le cas d'un réseau classique | 48 |
| 3.13 | Architecture de consommation inter-zones..... | 50 |
| 3.14 | La consommation énergétique d'un appareil dans un foyer | 52 |
| 3.15 | La consommation au niveau d'une zone..... | 53 |
| 3.16 | La consommation d'énergie en temps réel dans un foyer..... | 54 |
| 3.17 | La consommation entre le mini-serveur et l'appareil | 54 |
| 3.18 | Simulation d'un appareil intelligent..... | 55 |
| 4.1 | Itinéraire d'un agent mobile dans le réseau | 58 |
| 4.2 | Le modèle <i>e-Marketplace</i> [48]..... | 60 |
| 4.3 | Coopérative multi-agent de négociation [49]..... | 60 |
| 4.4 | Schéma décentralisé (MineAgeHops) [50] | 61 |
| 4.5 | Architecture MAgNET [51] | 63 |
| 4.6 | La relation entre le compteur intelligent et le <i>middleware</i> | 65 |
| 4.7 | Traitements des agents dans un routeur | 66 |
| 4.8 | Vecteur d'historique d'énergie utilisée par le foyer client..... | 67 |

| | | |
|------|--|----|
| 4.9 | Structure de paquet d'un agent mobile | 68 |
| 4.10 | Mécanisme de consommation et de vente d'énergie..... | 69 |
| 4.11 | Vecteur d'historique d'un service d'énergie produite localement | 70 |
| 4.12 | Vecteur de routage dans un agent mobile | 70 |
| 4.13 | L'agent mobile dans l'état initial | 71 |
| 4.14 | Structure de routage de l'agent mobile | 72 |
| 4.15 | Exemple de routage d'un agent mobile..... | 73 |
| 4.16 | Envoi d'un agent mobile à partir du site initial..... | 74 |
| 4.17 | Réception d'agent par un routeur..... | 75 |
| 4.18 | Réception des données par le foyer initial | 76 |
| 5.1 | Schéma du modèle simulé | 78 |
| 5.2 | La représentation de modèle simulé par le simulateur..... | 79 |
| 5.3 | Nombre d'agents sur la <i>File d'attente 1</i> dans le <i>Routeur 1</i> en fonction du temps | 80 |
| 5.4 | Nombre d'agents sur la <i>File d'attente 2</i> dans le <i>Routeur 1</i> en fonction du temps | 81 |
| 5.5 | Nombre d'agents sur la file d'attente dans un compteur intelligent en fct du temps.... | 81 |

RÉSUMÉ

L'évolution des réseaux électriques combinés avec les réseaux et les infrastructures des technologies de l'information et de la communication a donné lieu à une nouvelle génération de réseaux appelés *Réseaux électriques intelligents (Smart grids)*. Ces réseaux ont des exigences relativement aux architectures de communication et des services. Cette génération de réseaux requiert également la mise en œuvre de services pour leur monitoring et leur surveillance.

Avec la venue des compteurs intelligents, il devient possible de faire communiquer les foyers avec un centre de contrôle et de monitoring, ce qui permet de gérer et de contrôler à distance les appareils se trouvant dans le foyer. Les réseaux intelligents offrent également la possibilité d'intégrer des énergies alternatives et renouvelables comme l'énergie solaire ou l'énergie éolienne. Tout foyer connecté au réseau peut non seulement consommer de l'énergie provenant d'un réseau électrique comme celui d'Hydro-Québec, mais également produire de l'énergie et la rendre disponible aux autres foyers.

Dans ce mémoire, nous nous sommes intéressés à plusieurs de ces aspects et nous formulons des propositions pour répondre aux nouveaux besoins créés par ces réseaux. Nous avons proposé des infrastructures de communication impliquant des réseaux sans fil et une architecture permettant de connecter les différents éléments du réseau et les équipements des foyers (appareils électriques, appareils électroménagers, etc.). Nous avons proposé plusieurs services pour effectuer le contrôle et le monitoring du réseau et des équipements des foyers à travers les compteurs intelligents. À cet effet, nous avons utilisé des services Web. Nous avons proposé une procédure d'ordonnancement des appareils électriques et électroménagers des foyers en vue de gérer leurs consommations et faire face aux situations de panne, notamment dans les chutes de tension. Nous avons également développé des services et une application pour donner accès à ces services à travers des appareils portables, notamment les téléphones mobiles intelligents.

Pour permettre aux fournisseurs d'énergie de mettre à la disposition des usagers du réseau le surplus d'énergie qu'ils produisent, nous avons mis sur pied une infrastructure et des mécanismes d'échange pour la négociation de l'offre et de la demande d'énergie. Ce mécanisme utilise les agents mobiles. Ces agents sont des programmes qui voyagent à travers le réseau grâce à un protocole de routage spécifique en vue de trouver les meilleurs fournisseurs qui répondent aux demandes des foyers qui ont un besoin en énergie. Ce protocole de routage est évalué par simulation, et les résultats sont présentés et discutés dans ce mémoire, démontrant ainsi des bonnes performances.

Mots clés : réseaux électriques intelligents, compteurs intelligents, monitoring et contrôle de la distribution d'énergie, gestion de la consommation électrique, ordonnancement, communication sans fil, agents mobiles.

X

INTRODUCTION

0.1 – Introduction générale

Les réseaux électriques sont omniprésents dans notre vie quotidienne. Ils offrent de l'énergie en continu provenant de sources de production diverses et variées comme les barrages hydroélectriques, les centrales nucléaires, etc. Ils doivent fournir une électricité de qualité en maintenant l'amplitude et la fréquence de l'onde de tension à des valeurs acceptables par rapport aux normes de l'industrie.

La tension électrique sortie des générateurs est distribuée dans le réseau afin qu'elle puisse servir chaque point du territoire couvert par le réseau. Des transformateurs branchés directement à la sortie des alternateurs produiront cette tension. Celle-ci est ensuite transformée et transportée sur de grandes distances par des lignes de haute tension afin d'alimenter les différentes régions et différents foyers [1]. Ce mécanisme de distribution est représenté par la figure 1.

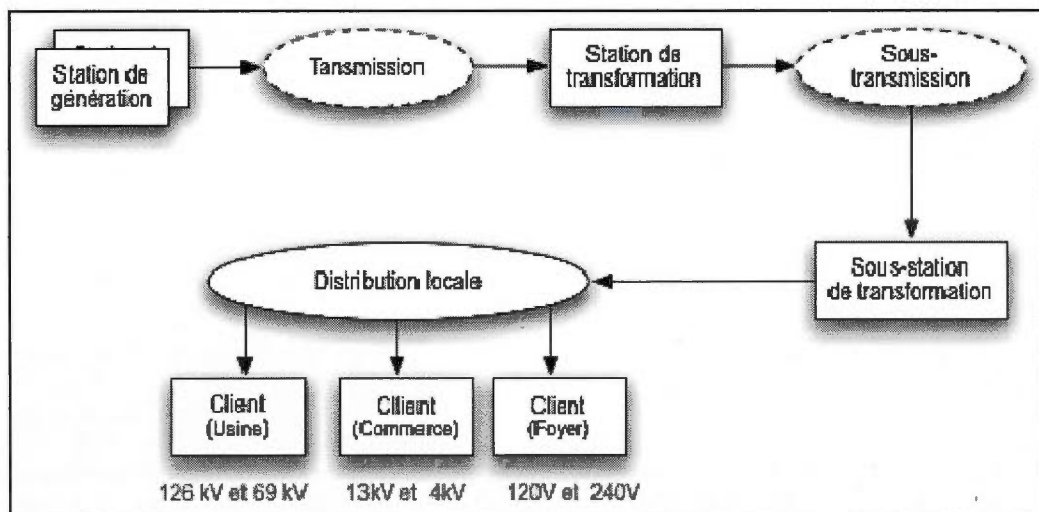


Figure 0.1 – Schéma de production, du transport et de la distribution d'énergie

Le contrôle et le monitoring en temps réel de cette distribution au niveau des régions et de chaque foyer nécessitent une architecture permettant d'accéder à chaque point du réseau et ce, dans le but de collecter les informations nécessaires pour exécuter ces tâches de soutien comme l'augmentation de tension, la coupure d'alimentation lorsque nécessaire, etc.

Dans les réseaux actuels, ces opérations de contrôle et de monitoring sont effectuées en envoyant des agents sur place pour exécuter des tâches comme le prélèvement de la consommation des foyers, la suspension de service, etc.

En vue d'automatiser ces tâches, plusieurs recherches ont été effectuées afin d'implémenter des mécanismes qui permettent de bien distribuer l'énergie et décentraliser ses ressources. Ce type de distribution permet également d'intégrer les énergies renouvelables et alternatives. Par exemple, l'énergie solaire captée par des panneaux solaires permet de récupérer une partie de l'énergie du rayonnement solaire pour la convertir en énergie utilisable. Les progrès technologiques dans ce domaine permettent de l'utiliser au niveau d'un simple foyer.

L'utilisation de ces énergies alternatives au niveau des foyers permettrait donc de diminuer la centralisation des ressources énergétiques, de diminuer la dépendance à l'égard des opérateurs et de réduire les coûts d'acquisition. Un simple client dans cette situation peut également devenir producteur d'énergie et offrir ses services à l'échelle du réseau de distribution [4].

Étant donné les besoins énergétiques, il serait impossible d'avoir une production énergétique uniquement issue des énergies alternatives, sauf dans certains cas peu nombreux dans des pays disposant d'un fort potentiel solaire ou hydroélectrique. De plus, ces énergies renouvelables font face à des défis de stockage et de coûts d'investissement.

La multitude des ressources énergétiques et l'augmentation du nombre d'utilisateurs et de leur consommation ont obligé à introduire un autre type de réseaux électriques appelés *Réseaux électriques intelligents (Smart grids)*. Un réseau intelligent permet d'intégrer de l'électricité provenant des fournisseurs aux consommateurs en utilisant les technologies de

l'information et de la communication dans le but d'économiser l'énergie, réduire les coûts et améliorer sa fiabilité et sa disponibilité [5].

Avec l'accroissement de la demande d'énergie et la demande pour augmenter les capacités de production, le système de transmission et de distribution doit inévitablement se préparer à distribuer de l'électricité plus efficacement tout en gérant au mieux les pertes. Les méthodes d'échange et des protocoles de communication adéquats doivent pouvoir aider à atteindre ces objectifs. De plus, les contraintes environnementales obligent les services publics à améliorer la façon dont ils gèrent les moyens de transport et de distribution afin de répondre aux besoins de leurs clients [6].

Un réseau intelligent détecte et traite les problèmes dans les systèmes de distribution avant qu'ils n'affectent les services. Il incorpore des systèmes de mesure et de communication rapides pour centraliser et diagnostiquer ces systèmes pour les ramener rapidement à un état stable après des interruptions ou des perturbations [7].

L'évolution des technologies permet d'intégrer de nouveaux éléments matériels et logiciels au niveau de foyers afin de faciliter les tâches de commandes et de *monitoring*. Une nouvelle génération de compteurs sera installée au niveau des foyers [40]. Il s'agit de compteurs intelligents (*Smart meters*) [8]. Ces compteurs communiquent en temps réel avec des centres de contrôle. Ils permettent de mesurer la consommation énergétique des foyers ou des appareils électriques et électroménagers auxquels ils sont connectés [41]. Ils pourront ainsi fournir un historique de la consommation énergétique d'un abonné d'une manière précise durant une période de temps donnée [9].

Plusieurs aspects doivent être considérés à ce niveau. À titre d'exemple, nous citons :

- La création d'un système de communication entre les compteurs [10]
- La mise en œuvre de protocoles de communication au niveau des compteurs afin de faciliter l'accès aux données qui y sont stockées, ou exécuter des opérations techniques à distance [11].

- La mise en œuvre d'un ensemble de services qui gèrent les informations de consommation et les rendent accessibles à distance, notamment à l'aide d'appareils mobiles [12]; et
- La mise en œuvre de procédures de négociation qui permet de choisir un bon fournisseur d'énergie dans un réseau ou une région formée d'un ensemble de foyers comprenant des fournisseurs et des consommateurs [13].

0.2 – Problématiques

Dans un réseau électrique intelligent, plusieurs défis se présentent au niveau de l'installation des systèmes de mesure et de captage, de l'infrastructure de transport de l'énergie, des mécanismes d'échange d'informations relatives à la consommation et des paradigmes de négociation pour l'offre et la demande de l'énergie [14].

L'installation d'un système électrique intelligent nous oblige à améliorer le comportement et la fonctionnalité des compteurs électriques classiques et des appareils électroménagers au niveau des foyers.

Dans notre travail, nous allons nous concentrer sur la problématique de l'installation des systèmes intelligents de contrôle de la consommation et des mécanismes de communication. Nous visons également à aborder certains aspects des mécanismes de négociation entre un consommateur et des fournisseurs d'énergie (alternative, par exemple). L'architecture de communication utilisée dans ce mécanisme doit être basée sur les technologies de communication comme les réseaux IP, les réseaux de téléphonie 3G et les réseaux sans fil [15].

0.3 – Le rôle des compteurs intelligents

Les compteurs intelligents permettent, entre autres, de gérer les réseaux électriques décentralisés avec des sources d'énergie diversifiées. La distribution et le contrôle de cette énergie doivent être gérés par un mécanisme intelligent [8].

Également, dans un réseau intelligent, étant donné qu'un simple client peut passer d'un statut de consommateur à celui de fournisseur d'énergie, le réseau doit donc permettre de gérer cette nouvelle situation.

L'utilisateur doit également pouvoir consulter en temps réel sa consommation énergétique en utilisant différents types de périphériques comme les ordinateurs portables et les mobiles pour communiquer avec ces compteurs.

L'évolution des technologies permet d'intégrer de nouveaux éléments matériels et logiciels dans un compteur, offrant ainsi la possibilité de capter les informations qui permet de contrôler la consommation énergétique d'un foyer en temps réel. De plus, ces compteurs doivent pouvoir communiquer entre eux, collecter la consommation énergétique des foyers ou celles des appareils auxquelles ils sont connectés. Ils pourront ainsi fournir un historique de la consommation énergétique d'un foyer afin de la suivre et de l'étudier de manière précise. Parmi les solutions proposées, il y a la réalisation d'un système de communication au niveau d'un foyer qui gère le transfert des données entre les appareils intelligents et entre ces derniers et les compteurs intelligents [9].

Au niveau des foyers, différents types d'appareils électroménagers munis de liens appropriés permettent de communiquer avec un compteur intelligent grâce à une liaison sans fil ou filaire. Dans ce contexte, nous visons à considérer plusieurs aspects, notamment :

- La mise sur pied d'un système de communication sans fil entre les compteurs pour leur permettre de communiquer entre eux et avec les appareils installés dans le foyer. Plusieurs types de technologies peuvent être utilisés à ce niveau (*ZigBee*, par exemple) [10]. Ces compteurs doivent également permettre de communiquer avec des réseaux externes. Ainsi, un centre de contrôle peut prendre les prélèvements de consommation sur un compteur à distance. Le client peut aussi consulter en temps réel sa consommation d'énergie.
- La réalisation d'un protocole de communication au niveau de ces compteurs pour gérer les cas d'urgence (par exemple, une chute de tension dans le réseau) [11]. Ce

protocole gère le comportement des appareils électroménagers installés dans un foyer afin d'éviter les perturbations des charges.

- L'introduction d'un mécanisme d'échange entre les compteurs intelligents dans différentes zones du réseau afin de gérer les offres et les demandes d'énergie. Cette offre pourra être fournie par plusieurs fournisseurs qui utilisent divers générateurs d'énergie [13]. Avec l'utilisation des énergies alternatives au niveau des foyers, un client dans le réseau peut vendre l'énergie produite localement. Cette production alimente son foyer, et en cas de surplus, elle sera injectée dans le réseau afin qu'un autre foyer puisse l'utiliser. Pour cela, ce mécanisme utilisera une procédure de gestion de l'offre et de la demande pour effectuer ces échanges. Ce mécanisme devrait être supporté par une procédure de négociation.
- La mise en œuvre d'un ensemble de services au niveau des compteurs intelligents afin de gérer les données qui représentent la consommation et les rendre accessibles à distance.

Mais à ce niveau, les problématiques sont diverses:

- Trouver un protocole standard de communication fiable entre les appareils électroménagers et les compteurs intelligents. Plusieurs fabricants d'appareils électroménagers travaillent pour élaborer ces standards afin de permettre à ces appareils de communiquer avec n'importe quel type de compteur intelligent [16].
- Trouver des solutions pour réduire le taux des collisions lors de la transmission des données par les appareils électroménagers et les compteurs intelligents.
- Établir des mécanismes permettant de mieux gérer et de contrôler la consommation énergétique des appareils électroménagers. Grâce à ces mécanismes, les compteurs intelligents peuvent minimiser le taux de gaspillage d'énergie dans les foyers, et aider à éviter le problème de chute de tension dans le réseau.

0.4 – L'intégration des énergies alternatives

Les recherches et les technologies courantes se basent sur la création de systèmes électriques qui produisent l'électricité tout en visant de lutter contre le réchauffement climatique et la pollution [21].

Dans un réseau électrique intelligent, la création d'énergie ne se limite pas aux seuls producteurs classiques (comme Hydro-Québec). On peut obtenir de l'énergie alternative supplémentaire et renouvelable créée par les utilisateurs qui deviennent producteurs et fournisseurs du réseau [3]. Ils peuvent donc vendre ce qu'ils produisent aux autres clients.

L'une des problématiques est de trouver un mécanisme automatique d'échange afin de permettre aux consommateurs de déterminer les bons fournisseurs d'énergie qui existent dans le réseau, ce qui nécessiterait un mécanisme de négociation et de choix parmi les fournisseurs qui présentent les meilleures offres [22].

Un fournisseur d'énergie peut être exposé à des problèmes techniques comme des coupures et des chutes de tension. Sur ce plan, on doit pouvoir trouver un mécanisme informatisé de prise en charge entre générateurs d'énergie afin de trouver rapidement des sources alternatives de courant [23].

0.5 – Objectifs

Un réseau intelligent est essentiel pour résoudre certains des problèmes exposés ci-haut. Par conséquent, nous visons les objectifs suivants:

- La mise sur pied des procédures permettant de maîtriser la consommation d'énergie dans un foyer ou une région. Un service de contrôle peut utiliser ces procédures pour lire en temps réel et à distance la consommation énergétique d'un foyer ainsi que la consommation globale d'énergie par une région (c'est-à-dire un groupe de foyers) en temps réel ou pour une période spécifique.

- La mise en place d'un outil de lecture de la consommation énergétique d'un foyer à partir d'appareils mobiles [18], ou par une consultation Web.
- La mise sur pied de procédures de gestion de la dissipation de puissance énergétique pour pallier les chutes de tension.
- La mise sur pied de procédures de courtage et d'achat/vente de l'énergie produite par les consommateurs.
- La mise sur pied d'une infrastructure de communication permettant d'effectuer ces échanges.
- La mise sur pied d'une infrastructure logicielle qui permet d'intégrer cette intelligence requise pour répondre aux nouveaux besoins qui sont apparus dans les réseaux intelligents.

0.6 – Méthodologie

La méthodologie appliquée est la suivante:

- La conception d'une architecture globale d'un réseau électrique intelligent sur laquelle se basera notre travail.
- La conception d'un schéma d'un compteur intelligent qui facilitera le fonctionnement de notre protocole de négociation.
- La définition d'une architecture de communication entre compteurs intelligents et les appareils électroménagers dans un foyer.
- La conception et l'implantation d'un algorithme d'ordonnancement permettant de faire fonctionner les appareils électroménagers tout en respectant des contraintes de consommation.
- La conception d'un protocole d'échanges pour supporter une procédure de négociation entre un vendeur d'énergie et un acheteur à travers le réseau en utilisant la technologie des agents mobiles.
- L'utilisation des agents mobiles pour exécuter les procédures d'échanges asynchrones de données en vue de procéder à la négociation de l'offre et la demande d'énergie.

0.7 – Organisation du mémoire

Après cette introduction, le reste du mémoire se présente comme suit:

- Le chapitre 1 donne quelques définitions propres aux réseaux électriques intelligents et un état de l'art dédié aux approches, méthodes et plateformes existantes pour la résolution de problèmes propres aux réseaux intelligents.
- Dans le chapitre 2, nous définissons une architecture globale d'échanges de données dans un réseau électrique basé sur deux voies de communication: *Wimax* et le Web. Nous mettons l'accent sur une proposition qui utilise les services Web. Nous nous intéressons également à évaluer par simulations le rendement d'un réseau électrique intelligent par rapport à un réseau classique et nous effectuons une comparaison des résultats entre les deux types de réseaux lors d'une chute de tension. Nous effectuons aussi des simulations de notre proposition de communication basée sur les services Web entre des foyers, un service de monitoring, et des clients.
- Le chapitre 3 expose les résultats de simulation du contrôle de la répartition des charges ainsi que les résultats de simulation de la consommation énergétique dans un réseau classique comparativement à la consommation énergétique dans un réseau électrique intelligent; le tout est suivi de l'étude de cas d'une chute de tension sur les deux types de réseaux et leurs réactions.
- Le chapitre 4 présente le protocole de négociation utilisant la technologie des agents mobiles entre un acheteur et des fournisseurs d'énergie.
- Le chapitre 5 contient la description d'une simulation du protocole de négociation *via* laquelle nous analysons le flux de données qui circulent au niveau des routeurs et des compteurs intelligents.
- À la fin de ce mémoire, nous tirons nos conclusions et nous présentons les travaux futurs.

CHAPITRE I

RÉSEAUX ÉLECTRIQUES INTELLIGENTS

1.1 – Définitions

Un réseau électrique intelligent peut être décrit comme étant une combinaison de systèmes qui s'articulent autour de trois types de systèmes (Figure 1.1):

- (1) Des systèmes de production d'énergie conventionnelle et renouvelable, lesquels forment l'ensemble des capacités de production électrique.
- (2) Des systèmes constitués des réseaux de distribution et de transport pilotés par l'offre et la demande d'énergie.
- (3) Des systèmes locaux d'intégration des énergies renouvelables et des systèmes de stockage, et même des véhicules électriques.

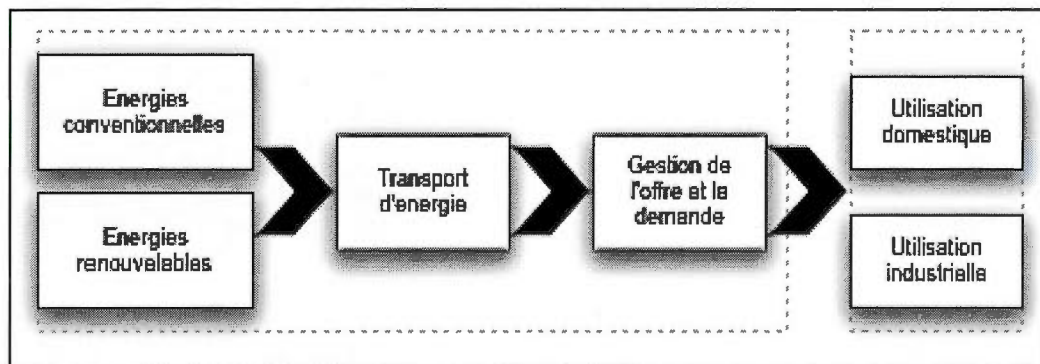


Figure 1.1 – Composantes d'un système de distribution intelligent

L'avènement de ce type de réseaux électriques a créé une harmonie entre l'environnement de production, de distribution et de gestion de l'énergie et celui des nouvelles technologies de l'information et des télécommunications.

L'infrastructure pour mettre en place un réseau électrique comportant trois niveaux:

- (a) Le niveau de capteurs et d'appareils évolués du réseau électrique.
- (b) Le niveau des applications pour la présentation des données et la prise de décisions.
- (c) Le niveau des réseaux de communication et plateformes d'intégration (Figure 1.2).

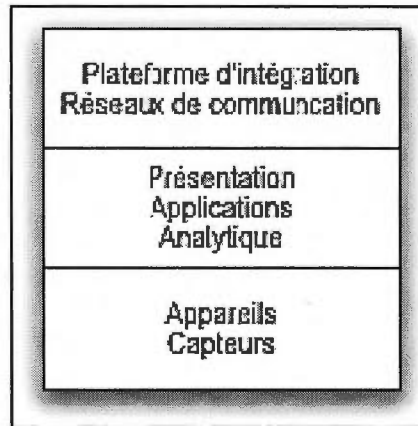


Figure 1.2 – Infrastructure d'un réseau électrique intelligent

1.2 – Gestion et distribution électrique dans un foyer

La production d'énergie dans ce modèle basée sur les réseaux intelligents est décentralisée. Elle comporte l'énergie produite classiquement et l'énergie renouvelable. Une zone est alimentée par des lignes de haute tension; cette énergie sera transformée afin d'alimenter le réseau local et en cas de surcharge, l'énergie de plus sera stockée pour utilisation ultérieure.

L'ajustement des tensions au niveau du réseau est géré par des modules de contrôle munis d'interfaces de communication sans fil, et téléguidés à distance par un centre de commande [24]. L'intégration des véhicules électriques dans le réseau a également été prise en considération dans ce type de réseaux. Des stations de recharge électrique sont installées sur plusieurs points dans le réseau. Ces stations alimentent ces véhicules dans des stations communes et même dans les foyers [24].

Il existe plusieurs modèles de distribution électrique comme (*Intelligent Distributed Autonomous Power System*) IDAPS [25]. Afin d'augmenter la fiabilité des services

électriques, des ressources distribuées d'énergies appelées (*Distributed energy resources*) DER ont été installées dans les artères de distribution électrique. IDAPS permet aux clients d'acheter l'électricité à partir de leurs voisins; le client utilisateur final aura donc le choix d'acheter une partie de leur électricité à partir des unités de DER disponibles localement si les offres présentent des prix compétitifs ou des caractéristiques intéressantes (comme l'énergie verte).

IDAPS utilise le système multi-agents. Chaque agent est mené par un service Web installé afin de partager ces données avec les autres agents sur le réseau (Figure 1.3).

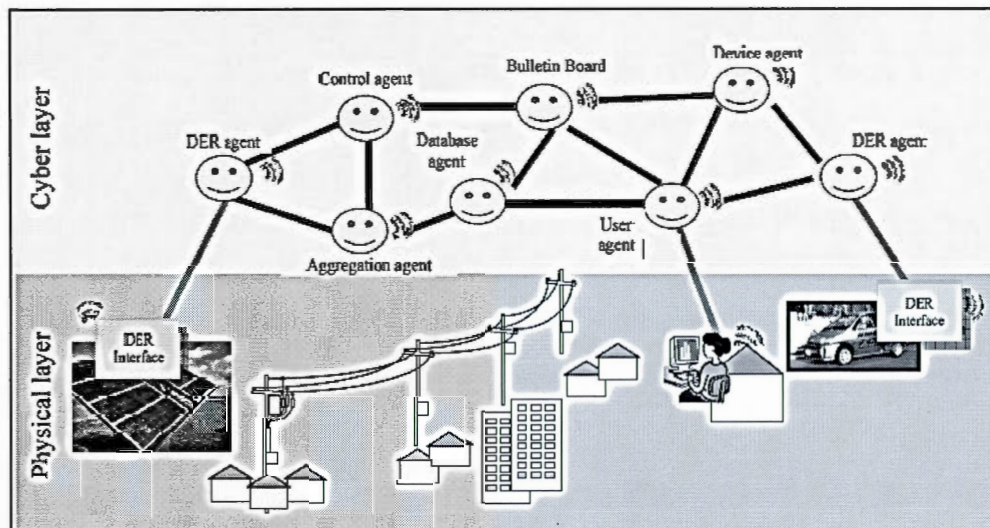


Figure 1.3 – Couches du système IDAPS [25]

Les agents dans IDAPS communiquent entre eux, et en cas de panne ou de chute de tension, plusieurs DER coopèrent afin de renforcer la charge. Le système (*Supervisory Control and Data Acquisition*) SCADA [2] est un système de gestion à distance pour traiter en temps réel un grand nombre de mesures et de contrôler à distance les installations techniques. Ce système peut être utilisé pour gérer les réseaux électriques, comme il peut aussi être utilisé sur d'autres applications ou domaines comme les canalisations de gaz et de pétrole, le transport de produits chimiques, etc.

La production décentralisée, ou (*Distributed Generation*) DG suscite de plus en plus d'intérêt et constitue une forme future de la production d'électricité. L'objectif est de créer un réseau électrique non centralisé afin que la production énergétique ne soit pas le monopole de quelques groupes ou entreprises.

1.3 – Gestion de l'offre et la demande

Étant donné que dans un réseau électrique intelligent, l'offre et la demande d'énergie électrique ne sont pas centralisées, un foyer peut vendre l'électricité à son voisin suivant une procédure d'offre et demande.

Sur ce sujet, plusieurs propositions avançant des solutions différentes en été publiées. Un système de distribution appelé (*Autonome Power System Area Demand*) ADAPS [26] est défini comme un segment du système de distribution connecté au système de transmission relié à la haute tension. Ce système maintient la stabilité et la fiabilité d'alimentation durant une longue période. Un module de contrôle et de distribution d'énergie est installé dans chacun des foyers et traite l'approvisionnement et la demande d'énergie, les batteries de stockage de la charge, en utilisant des techniques de communication (Figure 1.4).

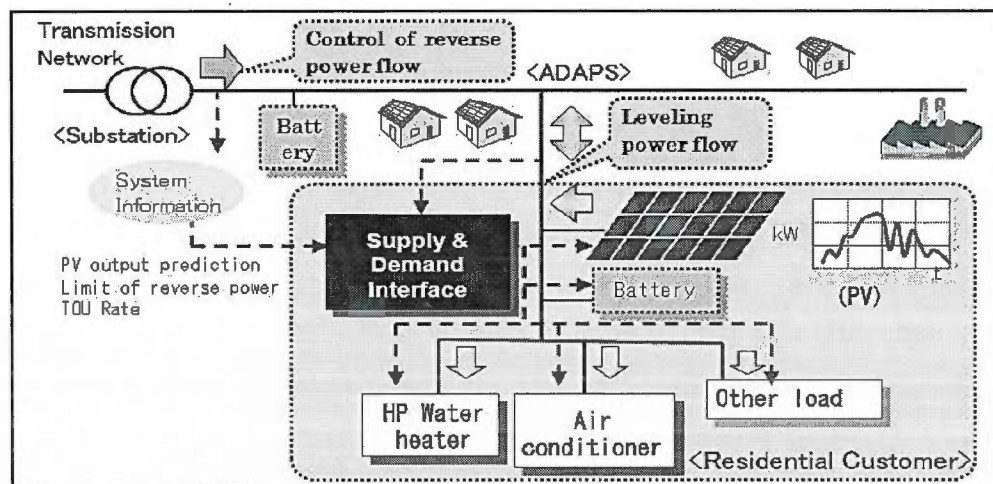


Figure 1.4 – Flux d'information dans ADAPS [25]

Dans ce modèle, un foyer peut injecter sa production énergétique sur le réseau, comme il peut recevoir, à travers un système d'information centralisé, des données qui seront traitées par la procédure d'offre et de demande.

Dans IDAPS, un mécanisme de gestion de l'offre et de la demande sur le réseau est également proposé. Si un utilisateur envoie une requête d'achat d'électricité, chaque agent dans le réseau reçoit une copie de la requête. Chaque agent enverra ses réponses vers ces destinataires et créera une copie des données envoyées dans une base de données centralisée. Cette base de données est accessible en temps réel par tous les agents (Figure 1.5).

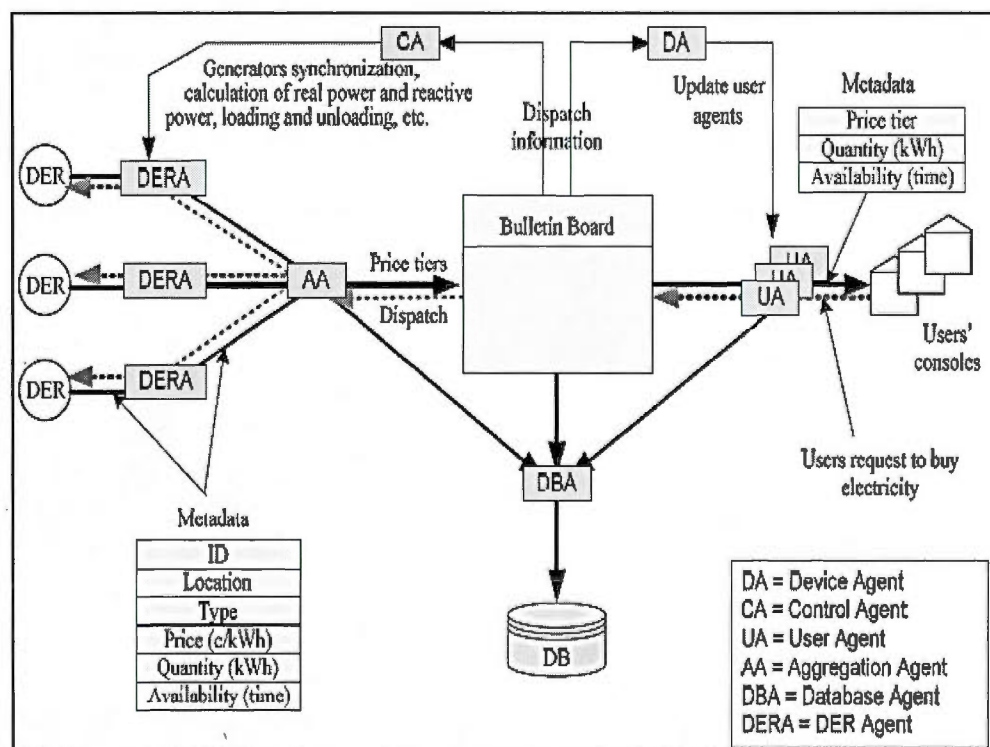


Figure 1.5 – Le cadre multi-agents dans une cellule IDAPS [25]

En comparaison avec notre proposition sur les procédures de négociation, dans IDAPS les agents de négociations communiquent entre eux pour échanger les informations des services publiés sur le réseau (achat/vente). Par contre, dans notre cas, un agent prendra les informations des niveaux des sites visités après une étape de négociation exécutée au niveau

du site fournisseur d'énergie. De plus, IDAPS est basé sur les services Web *via* des connexions filaires, alors que nous utilisons la transmission sans fil WIMAX.

1.4 – Accès et contrôle à distance

Grâce aux systèmes de contrôle à distance dans un réseau électrique intelligent, le client peut consulter sa consommation énergétique, ou même piloter ses appareils électriques et électroménagers. Dans ce domaine, nous citons quelques travaux similaires au nôtre.

Un système qui utilise des services Web est décrit [27]. Ce système comporte une passerelle résidentielle configurée qui possède un CPU, un module GSM, une interface réseau et un module de commutation. Toutes les composantes utilisées sont reliées entre elles dans un canal Ethernet intelligent au niveau du foyer (Figure 1.6).

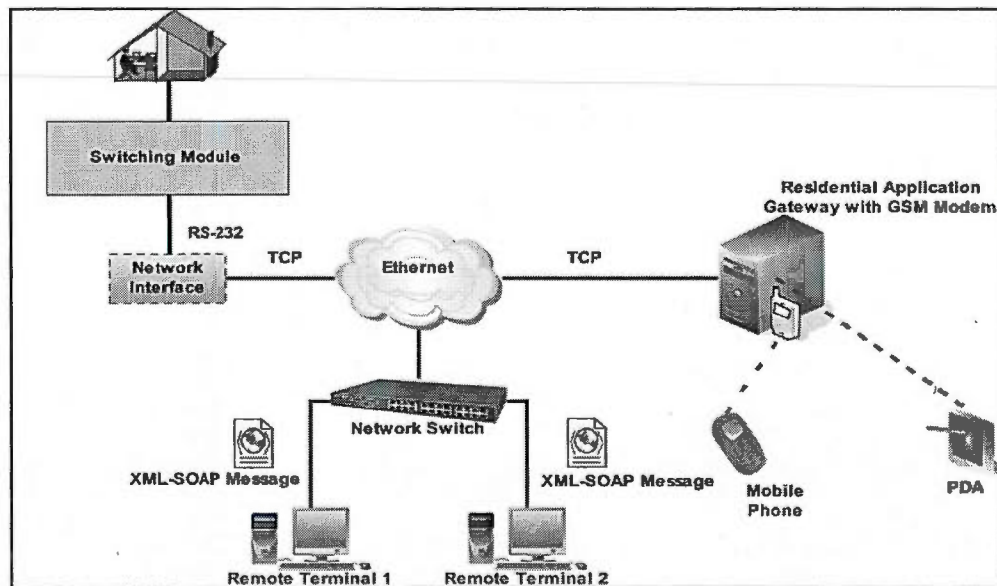


Figure 1.6 – Schéma architectural du système [27]

Un client envoie, à partir de son mobile, une commande vers la passerelle GSM (Gateway). Cette dernière reçoit cette requête et la transfère sur le réseau local comme un message (*Simple Object Access Protocol*) SOAP vers un terminal. Dans le terminal, il existe des

fonctions de commande déclarées comme des services. La requête reçue par le terminal amorce une fonction afin d'exécuter une tâche de commande qui sera envoyée vers une interface de type standard RS-232 vers un commutateur. Puisque chaque appareil électroménager dans le foyer est connecté à ce commutateur, le message de commande sera reçu par l'appareil auquel il est destiné afin qu'il change son état [27].

Dans le cadre de notre proposition, le service Web est représenté par les fonctions de commande qui se trouvent au niveau d'un serveur Web dans le foyer. L'accès à ces fonctions passe par l'intermédiaire d'un site Web installé sur un fournisseur externe qui utilise le service Web décrit par sa description (*Web Services Description Language*) WSDL. Le serveur est connecté à une interface *ZigBee*, et les appareils intelligents seront menés par le même type de connexion que celle installée sur cette interface. Cette solution offre une cadre plus souple d'accès aux services de commande du réseau.

La différence entre le modèle proposé dans [27] et notre proposition se résume dans les aspects suivants :

- L'accès aux données d'un compteur intelligent par un mobile dans le modèle précédent [27] passe par une connexion directe entre le mobile et serveur résidentiel à travers un module GSM. Par contre dans notre proposition, l'accès aux données d'un compteur passe à travers un routeur intermédiaire installé au niveau de chaque zone. Ce routeur est attaché à un service Web qui contacte les services Web résidentiels.
- Dans le modèle présenté dans l'article, un mobile utilise le réseau GSM pour contacter le serveur résidentiel à travers un module GSM. Par contre dans notre modèle la communication passe par l'utilisation de technologie WIMAX entre un mobile et un routeur, et un routeur et un autre.

L'architecture présentée dans [28] contient trois niveaux : (a) un dispositif de détection et d'enregistrement, (b) une plateforme de services Web (*Web Service Platform*) WSP et (c) une plateforme de services communs (*Common Service Platform*) CSP. Si le service de détection détecte un nouvel appareil, l'appareil sera enregistré par le CSP sur une base de données. Le CPS contient les interfaces d'accès aux services Web de chaque appareil. Tous les messages

se basent sur le protocole (*Simple Object Access Protocol*) SOAP. L'accès aux appareils électroménagers s'établit par la technologie (*Broadband Power Line Communication*) BPL. Dans ce modèle, les fonctions de commande sont installées sur la plateforme WSP. Ces dernières accèdent au CSP qui contient toutes les informations sur les appareils installés dans le foyer (Figure 1.7).

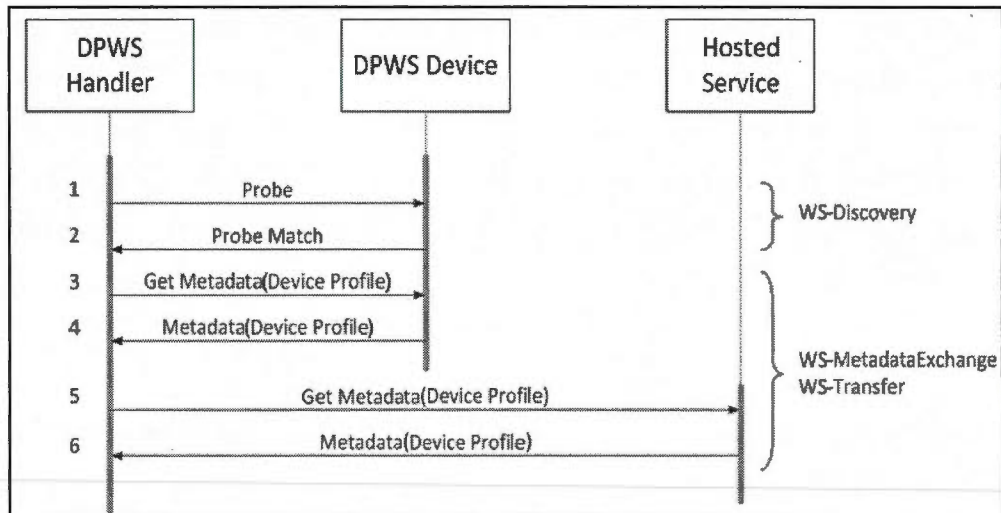


Figure 1.7 – Le flux de communication dans le service de détection [28]

Cette proposition est proche de la nôtre au niveau des services. Cependant, au lieu que le détecteur détecte un appareil, celui-ci envoie un message d'installation au serveur qui répond par un message d'acceptation, ce qui rend cette opération de détection automatique. De plus, l'usage de la technologie BPL oblige à modifier les installations électriques dans les foyers.

Le système *Energy Tracking* [29] propose un modèle de compteur intelligent qui utilise le service Web. Ce compteur contient :

- Les protocoles XML, HTTP, SMTP, SNTP, FTP, DHCP, DNS.
- Un service FTP client.
- SNTP pour la synchronisation du temps.
- L'authentification sur le serveur Web.

- Une base de données.
- Un serveur SOAP.

Un client peut accéder aux données du compteur grâce au service Web à travers un message courriel ou une requête FTP. Avec une application qui communique avec le compteur grâce à un service Web, un service de messagerie ou une connexion FTP, un client peut consulter sa consommation énergétique en temps réel. Comparé avec notre solution, ce modèle traite seulement les prélèvements électriques dans un foyer, et le traitement des prélèvements d'une zone (plusieurs foyers) sera effectué au niveau du centre de *monitoring*. Dans notre modèle, le traitement de données d'une zone sera effectué dans un autre serveur installé au niveau de la zone. De plus, dans notre modèle, nous avons ajouté une option où un client peut piloter un appareil électroménager à distance.

1.5 – Plateformes industrielles

Dans l'industrie, plusieurs constructeurs d'équipements et de logiciels ont proposé des environnements pour modéliser, gérer et surveiller les réseaux électriques. Nous citerons trois exemples de ce type de solutions.

Cisco Smart Grid est une combinaison de produits et de services dont l'architecture globale comporte quatre domaines principaux [30]:

- (1) Création, transmission, distribution et gestion de la production (centralisée ou décentralisée) d'énergie.
- (2) Gestion de la consommation au niveau des foyers, des industries et des commerces.
- (3) Gestion du réseau électrique : installations et maintenance des éléments du réseau.
- (4) *Monitoring* : facturation, gestion des données, etc.

Cette proposition est représentée par la figure 1.8 :

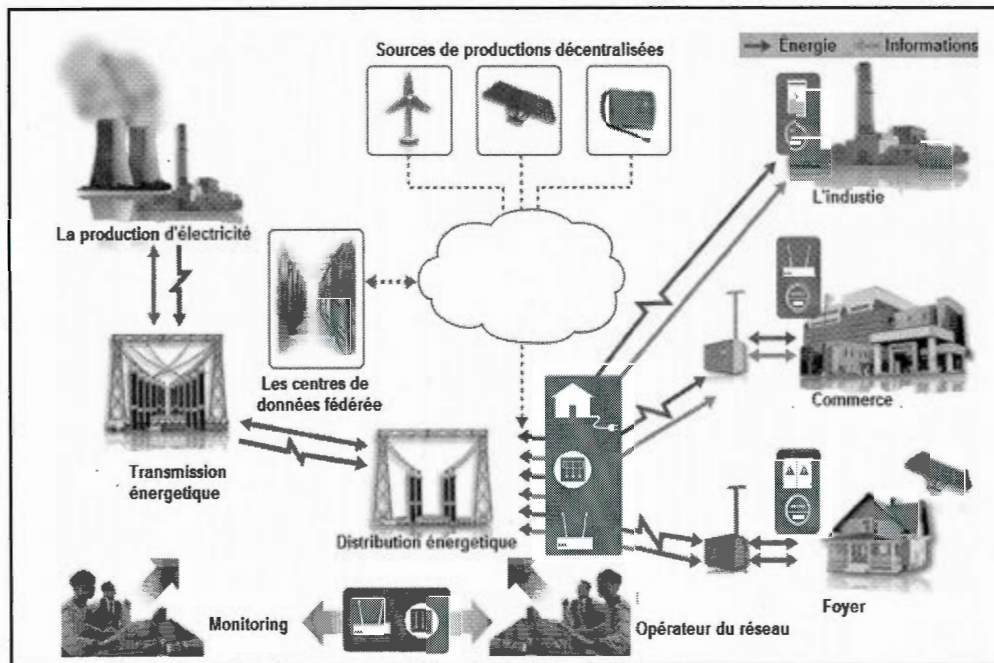


Figure 1.8 – La solution du réseau proposée par CISCO [30]

Les services fournis aux entreprises sont représentés par six niveaux :

- Le *Niveau 1* regroupe l'infrastructure d'énergie comme le câblage, les capteurs, les transformateurs, les connecteurs, les compteurs électriques, les unités de stockages, etc.
- Le *Niveau 2* représente les différents types de communications (*WiMax*, fibre optique, cellulaire, *Wifi*, *ZigBee*, etc.).
- Le *Niveau 3* représente la plateforme informatique de stockage de données.
- Le *Niveau 4* représente les systèmes opérationnels comme l'infrastructure de comptage automatisé (par exemple, les informations géographiques), les informations de comptage électrique, les données personnelles des clients.
- Le *Niveau 5* contient les applications commerciales utilisant les communications de compteurs, les données sur l'énergie produite localement, etc.
- Le *Niveau 6* représente les services commerciaux CISCO.

Le modèle CISCO offre uniquement une architecture globale de distribution d'énergie dans une architecture centralisée. Nous avons mis l'accent sur l'aspect communication et proposons une plateforme de communication et de négociation qui utilise une source de données non centralisée.

Un modèle proposé par *Oracle Corporation* comporte quatre domaines représentés comme suit [31]:

- (1) *Technologie client (CT)* – Ce domaine comporte la production décentralisée, les énergies renouvelables, le stockage d'électricité, les appareils intelligents et les véhicules électriques hybrides et rechargeables.
- (2) *Technologie opérationnelle / électrique (OT)* – Ce domaine est représenté par le modèle de gestion SCADA, les types de télécommunications, les capteurs et les technologies de contrôle, les réseaux sans fil, le comptage énergétique, etc.
- (3) *Compteurs intelligents (SM)* – Ceux-ci permettent à un utilitaire de recueillir les données à distance à des intervalles rapprochés.
- (4) *Informations/Technologie de données (IT)* – Elle comporte la gestion d'énergie des consommateurs, l'affichage des données, les applications d'entreprise, etc.

Ce modèle contient sept parties qui représentent les différents aspects d'un réseau électrique intelligent [31]:

- (1) *Les rapports analytiques*, comme les rapports clients, les rapports des commandes, les rapports de distribution énergétique.
- (2) *Les interfaces d'applications client*, comme les canaux d'interaction client (ventes, marketing, etc.)
- (3) *Les opérations commerciales*, c'est-à-dire la marchandise (l'énergie), la facturation et la gestion de devis, la maintenance, et la distribution d'énergie.
- (4) *L'administration d'entreprises*, c'est-à-dire les chaînes d'approvisionnement, l'étude des projets, etc.

- (5) *Le middleware*, qui représente les données stockées, l'infrastructure des données et la compression des données; cette partie contient aussi la partie réseaux informatisés qui représente les applications réseaux et les données qui circulent.
- (6) *La sécurité des données et la sécurité physique* (humain, matériel).
- (7) *Les applications*, qui comportent les applications installées au niveau du matériel ainsi que les infrastructures sur lesquelles ces applications s'installent.

Oracle travaille en collaboration avec des services publics en Californie pour installer l'un des premiers *microgrids*: des réseaux électriques de petite taille conçus pour être utilisés en dehors de ces grandes installations. Ce *microgrid* est un environnement autonome d'électricité qui fonctionne au sein des grands services publics d'électricité. Il facilite un usage local ainsi que la production et l'intégration d'énergies renouvelables.

Le modèle dit *Modèle de Chine* propose un réseau électrique intelligent basé sur des réseaux de haute tension et une architecture multicouche. Il intègre les dernières technologies de détection, de communication, d'information et de contrôle [32]. Il comporte les éléments suivants :

- (1) *La création d'énergie*, qui se base sur l'énergie produite par les réacteurs hydrauliques, les réacteurs à base de charbon, l'énergie éolienne, l'énergie solaire et les réacteurs nucléaires.
- (2) *La technologie de transmission d'énergie*, qui se base sur une architecture de transmission électrique de haute tension qui comprend les transformateurs, les redresseurs, les fils électriques, etc.
- (3) *La gestion de la fiabilité*, qui représente les contrôleurs de flux électrique de haute puissance, l'utilisation d'énergie renouvelable et l'injection de cette dernière dans le réseau général.
- (4) *La politique des prix*, qui affère à la tarification.

CHAPITRE II

ARCHITECTURES DE COMMUNICATION

2.1 – Introduction

Dans les réseaux électriques actuels, le contrôle et le *monitoring* des services s'effectuent par des méthodes classiques comme les prélèvements sur place et l'accès direct aux points à examiner. Plusieurs recherches ont été menées dans ce domaine en vue de réaliser de meilleures architectures afin de permettre de contrôler un réseau à distance [33].

Le choix d'un bon *design* du réseau est un aspect important pour trouver la meilleure solution pouvant aider à résoudre les problèmes de communication et de distribution énergétique et de contrôle.

Dans ce cadre, nous proposons une architecture de contrôle et de monitoring à distance, permettant à un centre de contrôle et de *monitoring* de communiquer avec des compteurs intelligents installés au niveau des foyers et raccordés au réseau.

À travers un compteur intelligent installé dans un foyer, le propriétaire peut consulter sa consommation énergétique à distance grâce à son téléphone portable ou son ordinateur.

Notre solution se base sur les technologies sans fil telles comme *WiMax* (IEEE 802.16) [34] (Figure 2.1). Dans cette architecture, un compteur intelligent est relié à un réseau externe par une liaison sans fil *WiMax*. En même temps, ce compteur communique avec les appareils intelligents installés dans le foyer par une communication sans fil de type *ZigBee* [10] (voir plus loin la figure 2.5). Cette communication est gérée par un serveur local relié à ce compteur ou intégré dans celui-ci comme on le verra plus loin dans ce chapitre.

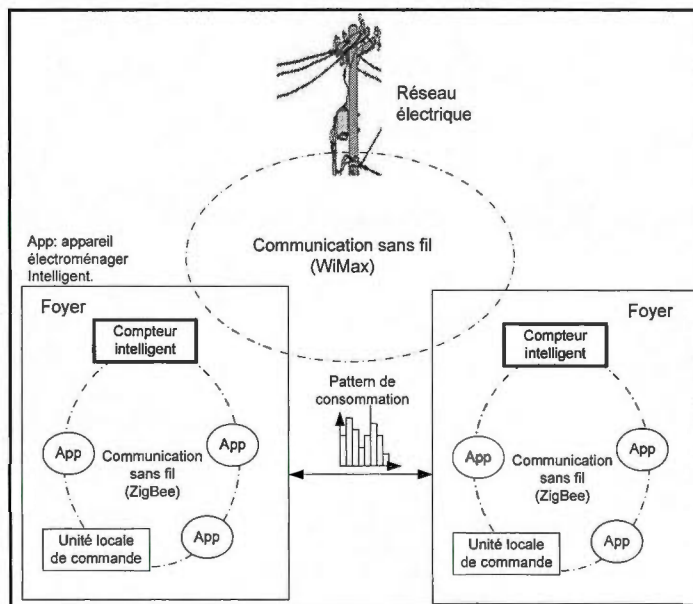


Figure 2.1 – Le modèle global d'un réseau électrique intelligent

Le serveur local est mené par un système d'exploitation *Windows*. Celui-ci comprend un serveur Web (en l'occurrence *Apache*). Une application serveur offrant des services Web utilisant le protocole d'échange SOAP a été installée sur ce serveur et contient plusieurs fonctions reliées à une base de données dans le compteur. Ces fonctions sont publiées par un fichier WSDL donnant des informations sur les services comme l'adresse du serveur, le port de communication, les fonctions publiées, etc. [17].

2.2 – Architecture de déploiement

Pour le déploiement de notre système, nous proposons une architecture de communication à deux niveaux (Figure 2.2):

- Le *Niveau région* représente la communication entre un compteur intelligent et un autre, ou entre un compteur intelligent et un appareil mobile, ou entre un compteur et un service de commande. On y utilise la technologie de communication *WiMax*.

- Le *Niveau foyer* représente la communication entre un compteur intelligent et les appareils électroménagers. On y utilise la technologie de communication *ZigBee* qui sera détaillée plus loin.

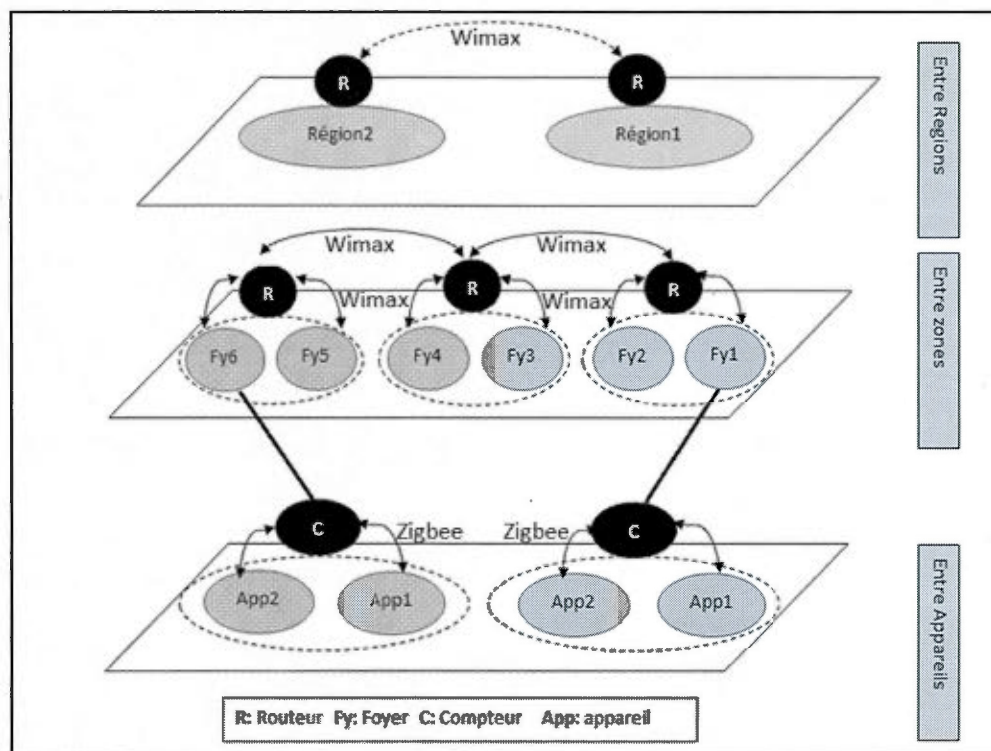


Figure 2.2 – Types de communications dans le réseau

Grâce à cette architecture, un centre de contrôle peut communiquer avec un compteur intelligent sur le premier niveau à travers des routeurs installés au niveau des régions et reliés par des liens *WiMax* ou par un accès direct au serveur installé dans le foyer.

Chaque zone regroupe plusieurs foyers et est gérée par un routeur qui contient trois tables de routage:

- Une table *Foyers* qui représente les foyers connectés à ce routeur.
- Une table *Fournisseurs* représentant les fournisseurs d'énergie dans la zone.
- Une table *Voisins* représentant les routeurs voisins.

Chaque routeur est représenté par une adresse IP unique dans le réseau global. Chaque routeur assure la communication et fait les liens entre les zones et les régions (Figure 2.3).

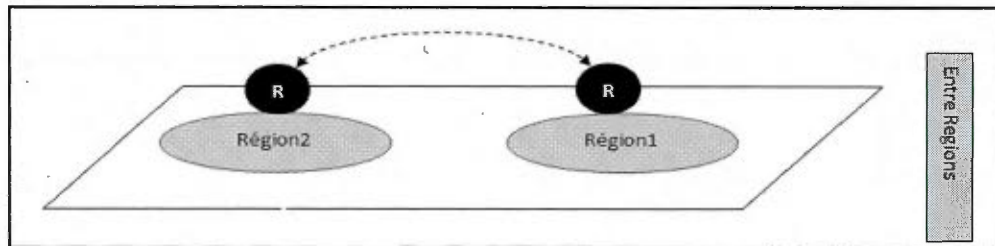


Figure 2.3 – La communication entre régions

Au niveau d'une région, la communication entre les compteurs intelligents se fait par l'intermédiaire des routeurs. Chaque routeur représentant une zone regroupe tous les compteurs connectés avec lui dans une table afin d'acheminer les communications (Figure 2.4).

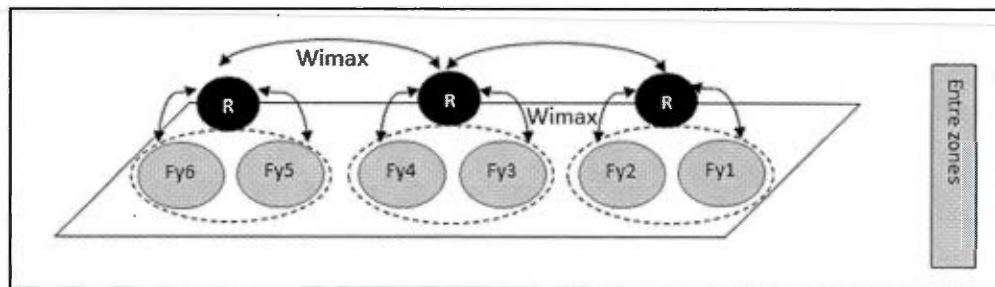


Figure 2.4 – La communication entre zones et foyers

Le choix de *WiMax* nous permet de créer un réseau *Mesh* ayant une portée pouvant atteindre 50 kilomètres [36]. De plus, les frais d'installation sont relativement réduits comparativement à des technologies équivalentes comme BPL [37].

Au niveau d'un foyer, on utilise une liaison *ZigBee*. Cette technologie est abordable et offre un niveau de fiabilité adéquat et une consommation électrique réduite. Un compteur relié *via* ce lien envoie des requêtes de commande aux appareils installés dans le foyer. Chaque requête est représentée par un message qui porte l'identification de l'appareil, le type de

message et les données à transférer. L'appareil électroménager possède une interface *ZigBee* qui peut communiquer avec le même type d'interface au niveau du compteur intelligent (Figure 2.5).

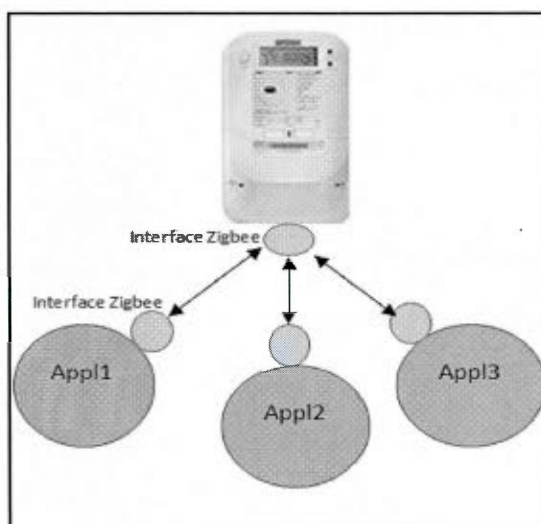


Figure 2.5 – Communication entre un compteur et les appareils électroménagers

2.3 – Les outils de déploiement

Pour implémenter cette architecture, plusieurs outils ont été développés et utilisés dans des versions améliorées.

2.3.1 – L'environnement du routeur

Un compteur intelligent communique avec un service de contrôle ou avec un autre compteur à travers une station de base. Cette station est représentée par un routeur d'acheminement. Chaque routeur utilise les trois tables citées précédemment.

Les tables de routage *Foyers* et *Fournisseurs* sont utilisées par le protocole de négociation de l'offre et de la demande d'énergie entre un client qui cherche un bon fournisseur d'énergie dans le réseau et les fournisseurs de ce service.

Chaque routeur possède une infrastructure qui facilite cette négociation en permettant d'exécuter et d'acheminer des agents mobiles chargés de cette négociation [38]. Il contient un environnement d'exécution de ces agents et un *middleware* tel que montré dans la figure 2.6. Le routeur regroupe ses tables de routage, un système d'exploitation (OS) et un *middleware* avec des modules de communication. Toutes les requêtes de communication, à l'exception de la procédure de négociation, s'exécutent dans la partie OS. La procédure de négociation s'exécute dans le *middleware*.

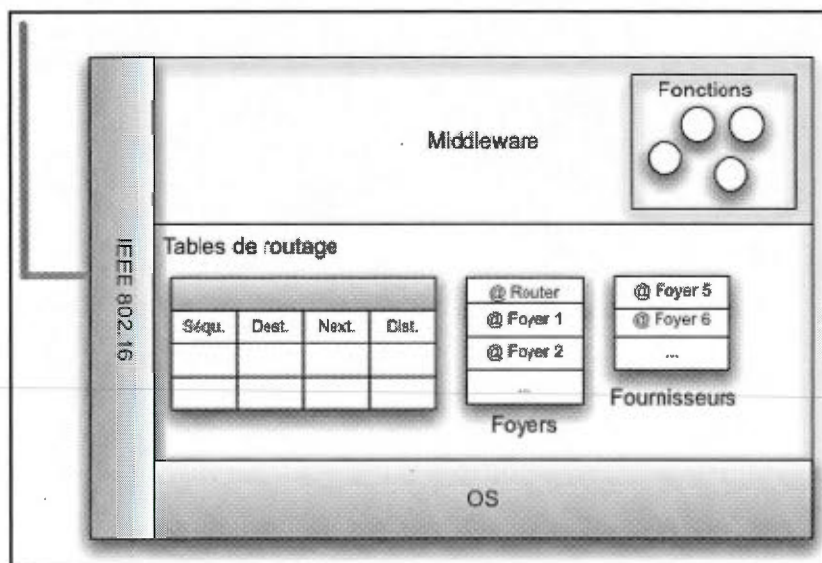


Figure 2.6 – Architecture d'un routeur

Le *middleware* constitue une plateforme d'exécution des agents mobiles chargés du mécanisme de négociation. Cette couche peut accéder à des fonctions du *middleware* pour le service de routage et pour l'accès aux bases de données ou les tables de routage. Parmi les fonctions de base, on a :

- Une fonction qui écoute sur un port de communication l'arrivée d'un agent mobile
- Une fonction qui exécute l'agent mobile
- Une fonction qui donne accès aux tables de routage et qui peut être utilisée par un agent mobile

- Une fonction qui joue le rôle de lien vers la base de données et qui peut être utilisée par l'agent mobile
- Une fonction qui duplique un agent mobile
- Une fonction qui retransmet l'agent vers un autre routeur ou vers un compteur intelligent
- Une fonction *DÉTRUIRE UN AGENT*.

Un routeur assure la communication entre les foyers d'une même zone et établit un lien vers les autres routeurs par l'utilisation des trois tables de routage citées précédemment :

- (1) La table *Voisins* contient les adresses IP des routeurs voisins et leurs caractéristiques de communication. Cette table est construite à l'aide de messages requête/réponse de type *Probe* [39] (Figure 2.7):

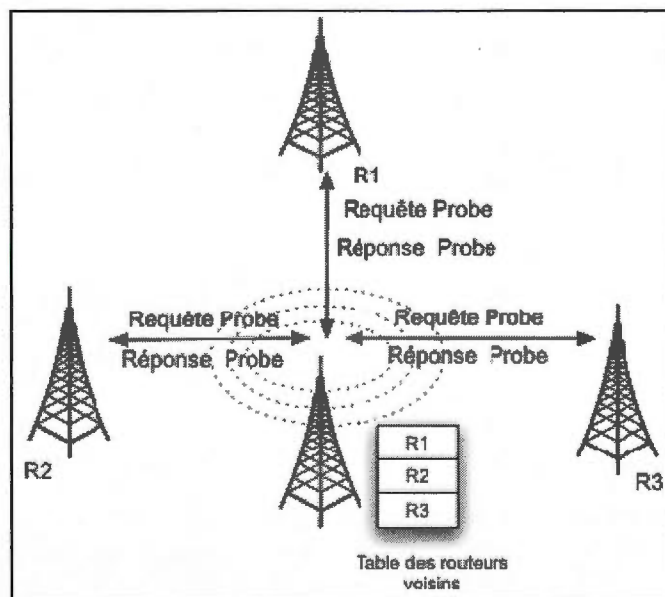


Figure 2.7 – Création de la table routeurs Voisin

- (2) Une table *Foyers* contient les adresses des compteurs intelligents connectés avec elle. Cette table possède la structure suivante (Figure 2.8):

| | | | |
|-----------|-----------|-----|-----------|
| @ Foyer 1 | @ Foyer 2 | ... | @ Foyer n |
|-----------|-----------|-----|-----------|

Figure 2.8 – Table de foyers connectés au routeur

- (3) La table *Fournisseurs* représente les foyers qui offrent le service de vente d'énergie dans une zone liée au routeur. Cette table prend aussi la forme d'un vecteur et contient l'adresse du routeur courant ainsi que les adresses des foyers qui fournissent l'énergie dans la zone (Figure 2.9):

| | | | | |
|-----------|-----------------|-----------------|-----|-----------------|
| @ Routeur | @ Fournisseur 1 | @ Fournisseur 2 | ... | @ Fournisseur m |
|-----------|-----------------|-----------------|-----|-----------------|

Figure 2.9 – Table des foyers qui fournissent l'énergie dans une zone

Les deux premières tables sont utilisées dans la procédure d'échange d'informations et d'exécution de commandes. La troisième table sera utilisée par le protocole de négociation mené par l'échange d'agents mobiles.

2.3.2 – Le compteur intelligent

Afin de permettre la création d'applications autour des réseaux intelligents, la technologie de ces compteurs a été améliorée de telle sorte qu'ils offrent un environnement riche en fonctionnalités. Ils ont aujourd'hui des caractéristiques qui facilitent leur intégration dans les grands systèmes de commande à distance.

Un compteur communique avec les services de contrôle et de commande et les appareils intelligents installés dans le foyer comme les postes de télévision, les machines à laver, les systèmes de chauffage, etc. (Figure 2.10).

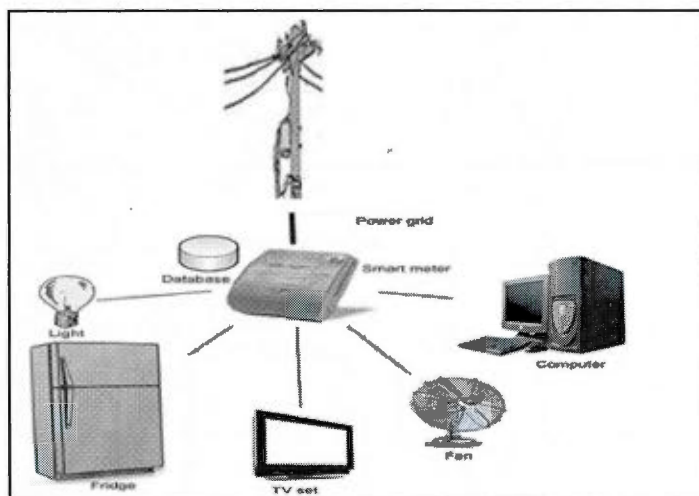


Figure 2.10 – Communication entre un compteur et les appareils électroménagers_2

Le rôle multiple du compteur nous oblige à lui adapter une architecture améliorée. Il contient une base de données gérée en temps réel qui sauvegarde l'historique d'utilisation d'énergie de chaque appareil. Il peut également gérer en temps réel le comportement de chaque appareil grâce à un système d'ordonnancement (décrit au chapitre 3) et une infrastructure de communication (représentée par la figure 2.11).

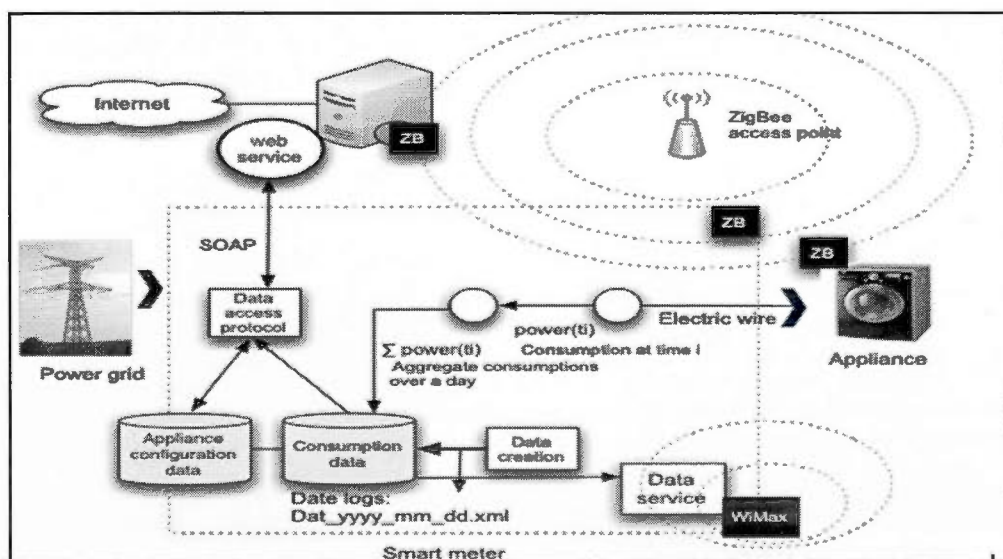


Figure 2.11 – Schéma d'un compteur intelligent

CHAPITRE III

SIMULATION DU CONTRÔLE DE LA RÉPARTITION DE CHARGE

3.1 – Environnement et paramètres de simulation

Dans ce chapitre, nous décrivons notre travail en évaluation de performances par simulation sur la répartition des charges. L'une des problématiques consiste à étudier le comportement de notre réseau dans le cas d'une chute de tension.

Une chute de tension peut se produire dans un réseau suite à une défaillance de celui-ci consécutive elle-même à une demande excédentaire d'énergie. Une chute de tension peut se caractériser par une diminution ou la disparition complète de la charge électrique, provoquant ainsi l'arrêt ou le mal fonctionnement des appareils. Afin d'éviter ces conséquences, il est important d'établir des algorithmes et des protocoles permettant de protéger certains appareils considérés prioritaires.

Pour étudier ce comportement et proposer des solutions à ce problème, nous avons mis sur pied des procédures et un environnement réseau permettant de :

- Effectuer un ordonnancement des appareils selon des critères comme leur durée de fonctionnement et leur niveau de priorité
- Faire face aux situations de chute de tension en utilisant l'ordonnanceur, lequel établit les règles de fonctionnement des appareils dans un foyer ou une région
- Permettre l'accès permanent et à distance aux infrastructures permettant de gérer la charge électrique dans un foyer ou dans une zone comprenant plusieurs foyers.

Les facteurs sur lesquels se base notre simulateur sont le niveau de priorité des appareils électroménagers installés dans les foyers, la quantité d'énergie consommée par chaque

appareil dans différents états, le taux d'utilisation de chaque appareil durant chaque période de la journée, la durée de la chute de tension et son intensité.

Notre solution est basée sur l'infrastructure réseau et les serveurs décrits dans la figure 2.2. Pour l'architecture des services, nous avons privilégié une approche orientée service basée sur les services Web.

3.2 – Simulation de la consommation

Avant de présenter notre travail de simulation, nous citons quelques travaux similaires dans ce sujet.

Un document décrit un simulateur d'une gestion intelligente de charges consommées par des appareils intelligents dans un foyer [44]. Dans cette étude, une enveloppe d'estimation de charges consommées par un appareil durant plusieurs sous-périodes sera déclarée manuellement dans chaque appareil électroménager dans un foyer. Cette enveloppe signifie que l'usage de charge par un appareil dans une période de temps ne doit pas dépasser le seuil déclaré. Avec ce mécanisme, la consommation énergétique sera gérée plus intelligemment en fonction de l'importance d'un appareil dans une période de temps. Par contre, dans notre modèle simulé, nous avons affecté une priorité à chaque appareil durant une période de temps par rapport aux autres appareils dans le foyer. De plus, au lieu d'étudier appareil par appareil, nous avons simulé plusieurs foyers en même temps où chaque foyer comporte plusieurs appareils gérés selon leurs priorités. Le compteur intelligent gère cette opération. La réaction des appareils intelligents dans un foyer en cas d'une chute de tension n'est pas prise en charge dans [44].

Dans [45], on présente une technique qui utilise les valeurs de puissance normalisées pour détecter l'état (marche/arrêt) des appareils, afin de bien gérer la distribution d'énergie dans un foyer (Figure 3.1).

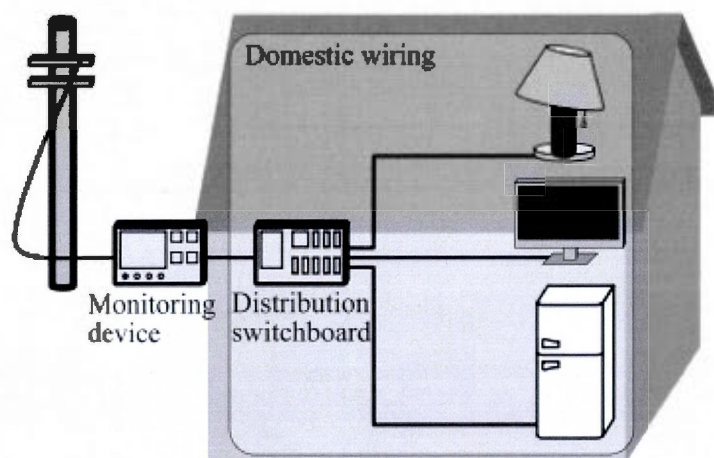


Figure 3.1– Schéma conceptuel d'un système de surveillance de la charge (NIALM) [45]

Cette technique donne de bons résultats, mais elle utilise comme variable la puissance réactive qui peut ne pas être disponible à partir des compteurs intelligents résidentiels. Cette technique utilise aussi un système personnalisé par une collecte de données qui peut être coûteux pour un déploiement à grande échelle.

Dans [46], un compteur intelligent contacte les différents appareils intelligents dans un foyer à travers des interfaces sans fil (*Zigbee*). Cette interface ramasse les informations qui représentent le comportement de chaque appareil installé dans ce foyer (Figure 3.2).

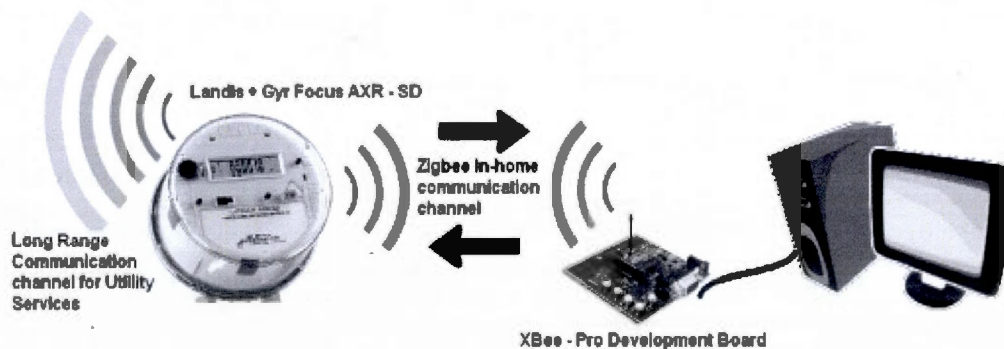


Figure 3.2 – Schéma global de solution proposé par Kuruppu [46]

L'algorithme de détection de comportement d'un appareil électroménager dans un foyer donne des intervalles de temps où l'appareil passe d'un état *OFF* à l'état *ON*, ou d'un état *ON* à l'état *OFF* (Figure 3.3).

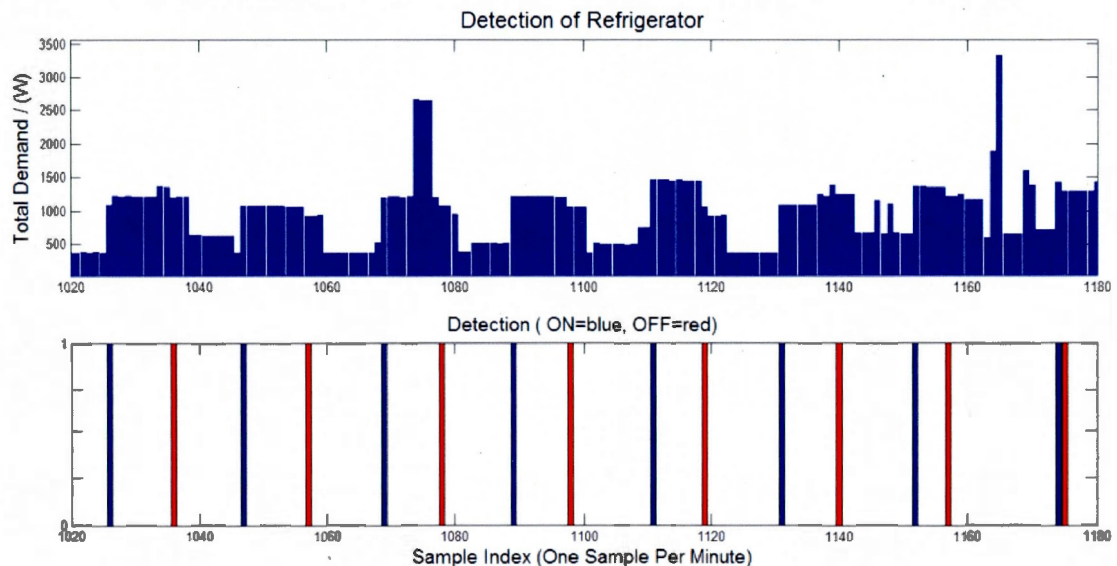


Figure 3.3 – Détection de comportement d'un appareil électroménager [46]

Le compteur intelligent collecte les données ramassées de chaque appareil et les utilise pour gérer le bon fonctionnement de ces derniers. Par exemple, il sélectionne les bons moments où un appareil peut fonctionner afin de minimiser les coûts.

D'un point de vue général, cette idée a donné de bons résultats par rapport aux autres travaux similaires. Elle donne des solutions pour minimiser les coûts, mais elle ne donne pas de solution pour gérer le comportement des appareils dans les cas particuliers de chute de tension par exemple.

Dans notre proposition qui utilise les priorités des appareils, un compteur intelligent gère plus intelligemment le comportement des appareils afin d'éviter une perturbation des fonctionnements des appareils dans un foyer. De plus, notre étude s'effectue sur plusieurs zones où chaque zone contient plusieurs foyers.

Le modèle de notre simulation dans un foyer utilise un algorithme qui fonctionne en temps réel durant le fonctionnement du compteur intelligent. Il considère que la journée est découpée en huit périodes de trois heures chacune. Chaque période est elle-même subdivisée

en soixante sous-périodes de trois minutes (Figure 3.4). Pour une journée, on a donc 480 sous-périodes. La durée de la chute de tension est de 405 minutes et l'intensité de cette chute est de 23000 watts.

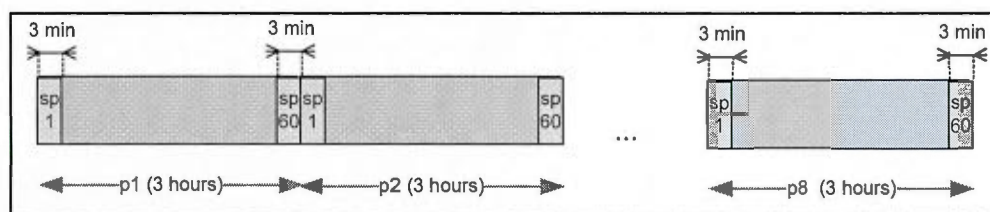


Figure 3.4 – Les périodes et les sous-périodes dans une journée

L'algorithme effectue un ordonnancement du fonctionnement des appareils en utilisant des priorités assignées à chaque appareil. L'algorithme utilise également des informations sur la consommation des appareils. Ces informations sont stockées dans un fichier XML dans une base de données du compteur. Ce fichier contient l'identité de chaque appareil (l'*ID* de l'appareil) suivie de sa consommation énergétique en état de marche (*POWER ON*), de sa consommation en état *OFF* (*POWER OFF*) et du taux d'utilisation de cet appareil durant chaque période avec le niveau de priorité de cet appareil durant cette période (Figure 3.5).

```

<? xml version="1.0" encoding="utf-8"?>
<appliances>
  <appliance>
    <name>App1</name>
    <power_on>115</power_on>
    <power_off>3</power_off>
    <perio1>
      <Use>12</Use>
      <Priorité>5</Priorité>
    </perio1>
    <perio2>
      <Use>11</Use>
      <Priorité>4</Priorité>
    </perio2>
    <perio3>
      <Use>16</Use>
      <Priorité>7</Priorité>
    </perio3>
    <perio4>
      <Use>18</Use>
      <Priorité>9</Priorité>
    </perio4>
    <perio5>
      <Use>19</Use>
      <Priorité>9</Priorité>
    </perio5>
    <perio6>
      <Use>15</Use>
      <Priorité>7</Priorité>
    </perio6>
    <perio7>
      <Use>13</Use>
      <Priorité>6</Priorité>
    </perio7>
    <perio8>
      <Use>16</Use>
      <Priorité>10</Priorité>
    </perio8>
  </appliance>
  ...
</appliances>

```

Figure 3.5 – Le fichier configuration d'un appareil intelligent

Chaque appareil électroménager installé au niveau d'un foyer doit posséder un niveau de priorité calculé en fonction des besoins de cet appareil durant une période donnée par rapport aux autres appareils.

Ce niveau de priorité comprend des valeurs entre 1 et 10. La valeur 10 sera affectée aux appareils les plus importants et les appareils seront classés en fonction de leur importance dans le foyer (comme suggéré dans la figure 3.6).

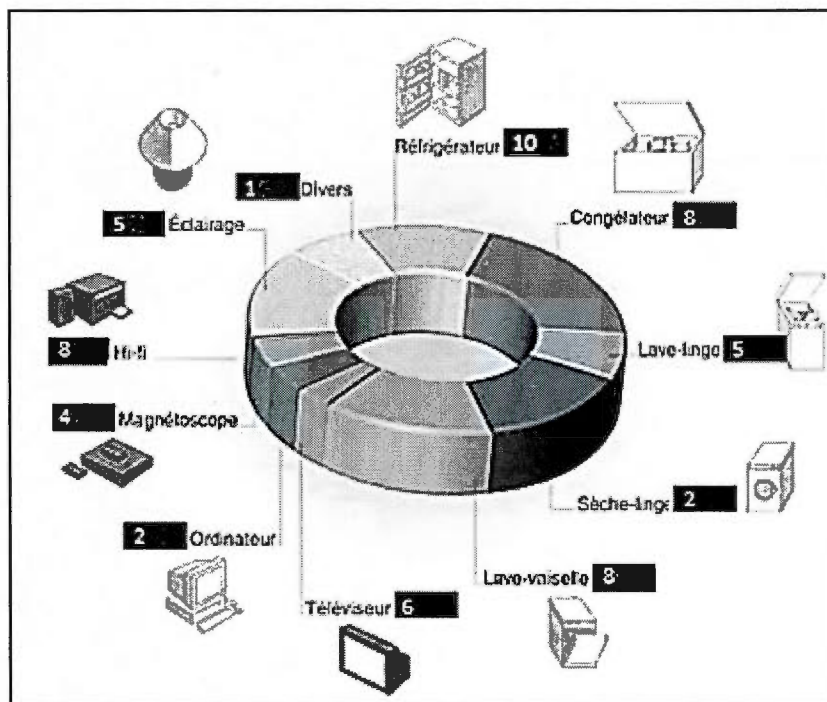


Figure 3.6 – Priorité des appareils dans un foyer [43]

En situation d'urgence, de chute de tension ou de dépassement du quota énergétique à consommer, ce compteur gère le fonctionnement des appareils en fonction de leurs priorités. Il envoie des requêtes aux appareils pour mettre ceux qui ont une priorité inférieure à l'état *OFF* afin d'éviter les conséquences néfastes des dites situations.

Pour simuler le comportement de chaque appareil dans une journée de fonctionnement, nous utilisons une fonction de randomisation qui génère une séquence de valeurs aléatoires pour définir les variations par rapport à l'utilisation d'un appareil.

Soit un appareil a à en état de marche (i.e., *ON*) et un pourcentage d'utilisation $n\%$ de cet appareil au cours de chaque sous-période, ce pourcentage varie entre $n-p_1$ et $n+p_2$, où p_1 et p_2 sont des nombres aléatoires qui prennent des valeurs oscillant entre $1-\varepsilon$ et $1+\varepsilon$. Elle représente

les faibles fluctuations de l'amplitude maximale 2ε au cours d'une sous-période donnée : elle est exprimée par $n*\rho$.

Dans un réseau électrique classique, la fonction utilisée pour mesurer la consommation énergétique d'un appareil a durant une période Δt est représentée ainsi :

$$Power_a(\Delta t) = power_ON_a * \Delta t + power_OFF_a * \Delta t$$

Et la consommation globale de tous les appareils dans le foyer durant cette période Δt s'exprime ainsi :

$$Power(\Delta t) = \sum_a Power_a(\Delta t)$$

Dans un réseau électrique intelligent, les consommations des appareils à l'état *OFF* sont mises à 0.

Avec l'utilisation des contrôleurs intelligents, chaque appareil peut obtenir de l'énergie à partir d'autres appareils afin de lui permettre de fonctionner. Les équations suivantes représentent la consommation énergétique dans le cas d'un réseau intelligent durant une période Δt :

$$Power_a(\Delta t) = \sum power_ON_a * \Delta t$$

$$Power(\Delta t) = \sum_a Power_a(\Delta t)$$

Dans le contexte d'un réseau intelligent, puisque seule la composante *POWER ON* est considérée pour chaque appareil, la consommation globale d'énergie d'un appareil durant une période Δt est représentée par la formule suivante :

$$SGC_{a,sp} = power_ON_a * P_{a,sp} * \rho * nz * nh$$

où $P_{a,sp}$ représente le pourcentage d'utilisation d'un appareil a durant une sous-période sp .
 La représentation des consommations énergétiques des appareils en fonction du temps dans une région est représentée sur la figure 3.7 :

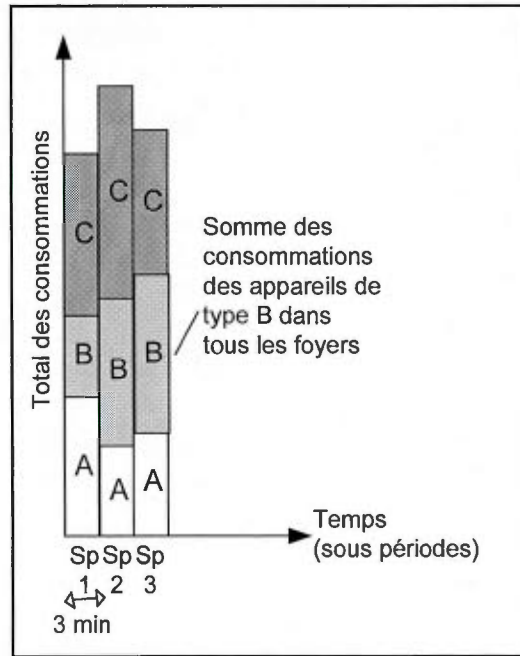


Figure 3.7 – Consommation énergétique des appareils dans une région

Étant donné que dans un réseau classique, nous devons considérer les états *POWER OFF* des appareils durant chaque période pour calculer l'énergie consommée par ces derniers, celle-ci est exprimée comme suit :

$$C_{a,sp} = SGC_{a,sp} + OFFER_{a,sp} * power_OFF_a * nz * nh$$

Pour un appareil donné et une période $Period_{a,sp}$, extraite à partir du fichier XML, $PUP_{a,sp}$ est égale à $Period_{sp}/100$:

$$OFFR_{a, sp} = 1 - PUP_{a, sp} * \rho$$

(Représentant les états *POWER OFF* d'un appareil *a* ; cette variable oscillant entre $1-\epsilon$ et $1+\epsilon$, avec une petite valeur de ϵ).

Pour la simulation, nous avons utilisé 4 types d'appareils différents dans les 5 zones. Dans chaque zone, on a 300 foyers (c'est-à-dire $N_z = 5$ et $N_h = 300$). Les données de consommation utilisées sont indiquées dans le tableau suivant [19] :

Tableau 3.1 – Données de consommation

| Nom d'appareil | Type | Consommation | Consommation dans une sous-période de 3 min. |
|-----------------|----------------------|--------------|--|
| Appareil1 | Four Classique ON | 2300 W/h | 115 W/3 min |
| Appareil1 STDBY | Four Classique OFF | 0 W/h | 0 W/3min |
| Appareil2 | Frigo 200 litres ON | 80 W/h | 4 W/3min |
| Appareil2 STDBY | Frigo 200 litres OFF | 0 W/h | 0 W/3min |
| Appareil3 ON | TV CRT 892cm ON | 120 W/h | 6 W/3min |
| Appareil3 STDBY | TV CRT 892cm OFF | 10 W/h | 0.5 W/3min |
| Appareil4 ON | Ordinateur /LCD ON | 75 W/h | 3.75 W/3min |
| Appareil4 STDBY | Ordinateur /LCD OFF | 3 W/h | 1 W/3min |

3.3 – Fichiers des données de consommation

La base de données de l'historique d'énergie utilisée par foyer est créée par l'algorithme décrit ci-dessous :

Variables utilisées :

- P_i représente une période i .
- P_r représente la période courante.
- Δt représente le temps courant.
- *ChargeGlobale* ($App, \Delta t$) représente l'énergie globale consommée durant une période de temps donnée Δt de 3 minutes et pour un appareil *App*.

- *GetPeriode(t)* donne la période correspondant au temps courant t .
- *ChargeGlobaleParPeiode(App)* représente l'énergie consommée par un appareil durant une période.

Une horloge envoie le temps courant à chaque instant. Ce temps est lu grâce à la fonction *LireTemps()* comme suit :

$$t = \text{LireTemps}()$$

Δt est calculée en fonction du temps courant t pour déterminer la période.

$$\Delta t = \text{getPeriode}(t);$$

À l'aide de Δt , on doit terminer quelle période P_i de 3 heures choisir parmi les 8 périodes de la journée. Le résultat est stocké dans la variable Pr .

$$\text{si } \Delta t > P_i \text{ et } \Delta t < P_{i+1}, \text{ alors } Pr = P_i; \text{ pour } i = 1 \dots 8$$

Ce test s'effectue seulement au début du traitement pour permettre au compteur de débiter à n'importe quelle sous-période de 3 minutes. La charge globale consommée par un appareil dans le foyer à chaque instant Δt est lue grâce à la fonction *ChargeGlobale(App, Δt)*.

La somme de valeurs de *ChargeGlobale(App, Δt)* représente la quantité d'énergie totale consommée par un appareil *App* durant une période Δt . Cette valeur est représentée par *ChargeGlobaleParPeriode* et est calculée par :

$$\text{ChargeGlobaleParPeriode} (App, P_i) = \sum_{\Delta t} \text{ChargeGlobale} (App, \Delta t) \text{ pour } \Delta t \text{ entre } P_i \text{ et } P_{i+1};$$

La valeur de *ChargeGlobaleParPeriode ()* est stockée dans un fichier appelé *GlobaleParPeriode.xml* dans la base de donnée du compteur à chaque période de la journée. Cette opération est représentée par le code (pour tout appareil *App* durant la période P_i) :

$$\text{EcrireXML} (jj_mm_aa.xml, \text{ChargeGlobaleParPeriode} (App, P_i))$$

Ainsi, le fichier créé contiendra la consommation des appareils durant toutes les périodes de la journée. Chaque journée aura donc son fichier, c'est-à-dire un total de 365 fichiers pour une année.

Dès que la variable *Temps* atteint le début d'une nouvelle période, la valeur de *ChargeGlobaleParPeriode* pour un appareil donné *App* est mise à 0 :

$$ChargeGlobaleParPeriode (App, Pi) = 0;$$

Le compteur peut commencer à n'importe quel moment dans une période, *ChargeGlobaleParPeriode()* comprenant uniquement les sous-périodes parcourues. La valeur se rétablit pour les périodes suivantes.

L'algorithme boucle durant le fonctionnement du compteur. Il débute par la lecture du temps courant tel qu'expliqué. Dès que l'algorithme aura traité toutes les périodes de journée, il créera un fichier XML *jj_mm_aa.xml* qui va contenir les données de la journée qui sont contenues temporairement dans le fichier *GlobaleParPeriod.xml*. Ce fichier journalier porte le nom qui contient le jour, le mois et l'année de l'opération d'écriture. Il est créé au début de chaque journée.

Si $\Delta t = P8$, on fait une copie du fichier *GlobaleParPeriod.xml* dans le fichier *jj_mm_aa.xml*. Ensuite, on vide le fichier *GlobaleParPeriod.xml* pour commencer une nouvelle journée.

3.4 – Coordination en cas de chute de tension

Nous avons étudié le comportement du réseau dans une situation de chute de tension. Une telle chute se produit si le réseau ne produit pas une quantité d'énergie permettant de supporter la consommation des appareils dans une zone ou un foyer. Dans le cas d'un réseau classique en situation d'insuffisance énergétique, certains appareils ne seront pas opérationnels. Dans le cas des réseaux intelligents, nous avons adopté une solution dans laquelle seuls les appareils les moins prioritaires ne devraient pas être opérationnels.

Le compteur intelligent du foyer gère le comportement des appareils électroménagers auquel il est relié et intervient dans une telle situation.

Pour cela, nous avons implémenté une procédure d'ordonnancement basée sur le niveau de priorité de ces appareils. Cette procédure envoie un message *GET* à tous les appareils qui ont priorité moindre (*Priorité 1*) pour les mettre en état *OFF*. Ces appareils répondent par un message *OK* et se mettent à l'état *OFF*. Si le problème persiste encore (c'est-à-dire si la demande d'énergie demeure supérieure à l'offre), le compteur envoie un deuxième message *GET* aux appareils qui ont une priorité 2. Le même scénario se répète jusqu'à ce qu'il considère que la demande d'énergie par les appareils dans le foyer est inférieure aux offres. Cette procédure est représentée par la figure 3.8.

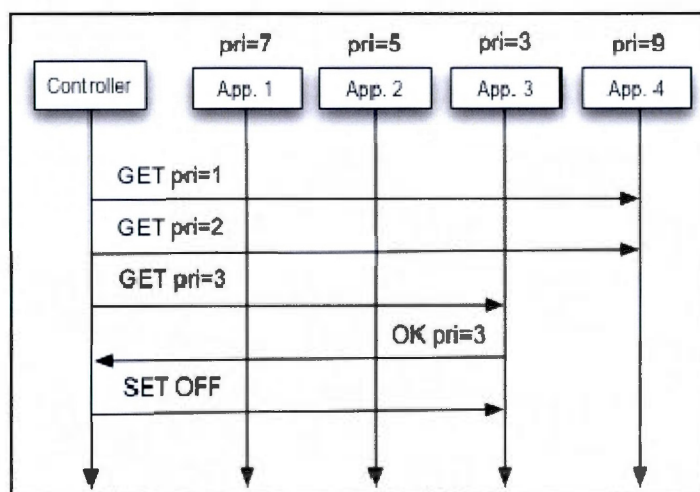


Figure 3.8 – Communication entre un compteur intelligent et les appareils électroménagers

3.4.1 – La consommation en état *OFF*

Les résultats des simulations sont représentés par les graphiques ci-dessous (Figure 3.9, Figure 3.10). Ces simulations représentent la différence entre un système qui prend en charge les consommations des appareils en état *OFF* et un système qui utilise les appareils électroménagers intelligents avec une consommation *OFF* égale à 0.

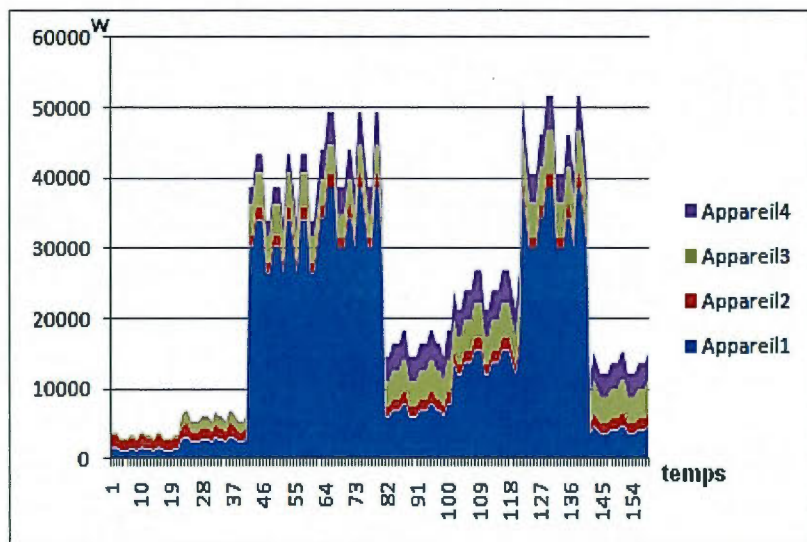


Figure 3.9 – La consommation des quatre appareils ;
le cas d'un réseau classique

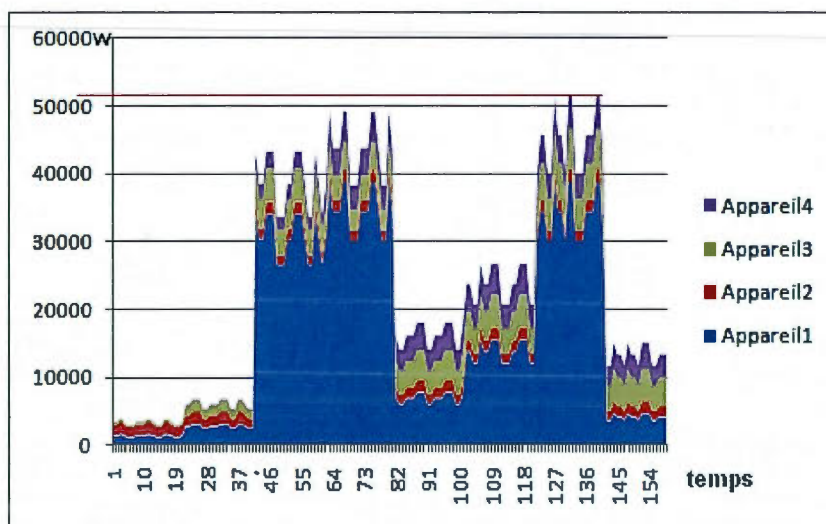


Figure 3.10 – La consommation des quatre appareils ;
le cas d'un réseau intelligent

Dans le cas d'un réseau électrique classique, l'ensemble des quatre appareils consomme 3 814 623,3755 w/jour. Dans ce cas, quelques appareils consomment une quantité d'énergie dans leur état *OFF* (Figure 3.9).

Dans le cas d'un réseau intelligent, la consommation est de 3 778 063,125 w/jour. La consommation de chaque appareil en état *OFF* est égale à 0 (Figure 3.10).

Considérant les coûts, la différence entre les deux cas est de 36 560 W/j pour les quatre appareils. Avec un prix approximatif 0,75 \$/kW [42], les calculs donnent un gain de 27 \$ pour 1 500 foyers / jour en faveur du réseau intelligent.

3.4.2 – Consommation en cas de chute de tension

Comme nous l'avons relaté précédemment, le compteur intelligent communique avec des appareils électroménagers munis d'interfaces intelligentes. En cas de chute de tension et pour éviter une perturbation du fonctionnement des appareils, ce compteur peut gérer la situation en se basant sur les priorités des appareils stockées dans sa base de données XML présentée dans la figure 3.5.

Les deux figures (Figure 3.11, Figure 3.12) montrent la différence entre le comportement d'un réseau intelligent et celui d'un réseau classique en cas d'une chute de tension.

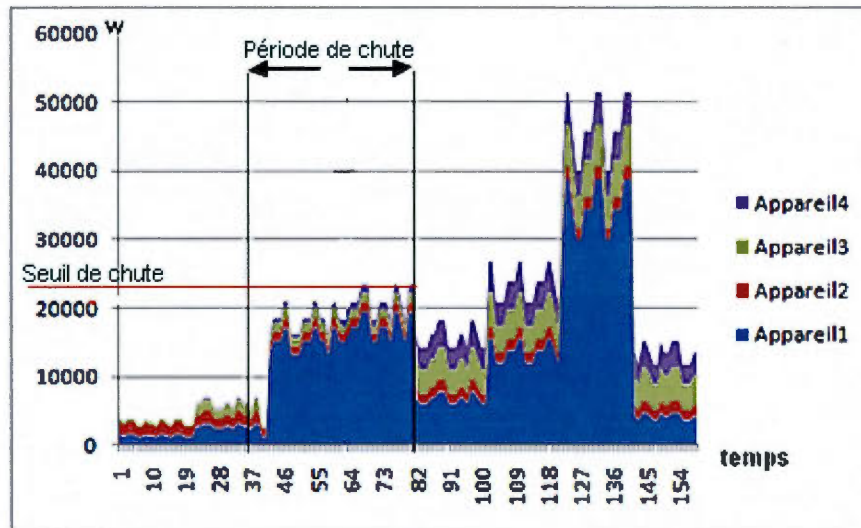


Figure 3.11– La consommation des appareils ; le cas d'un réseau intelligent

Avec un réseau électrique intelligent, le seuil détermine la valeur d'énergie consommée durant une période Δt qui ne doit pas être dépassée. Ce seuil est déterminé par le fournisseur. Cette valeur peut être également déterminée grâce à un contrat entre le foyer et le fournisseur. Habituellement, le fournisseur impose un seuil pour tous les foyers. Dans notre simulation, nous avons pris une valeur arbitraire de 23 000 W pour toute la zone de 1 500 foyers.

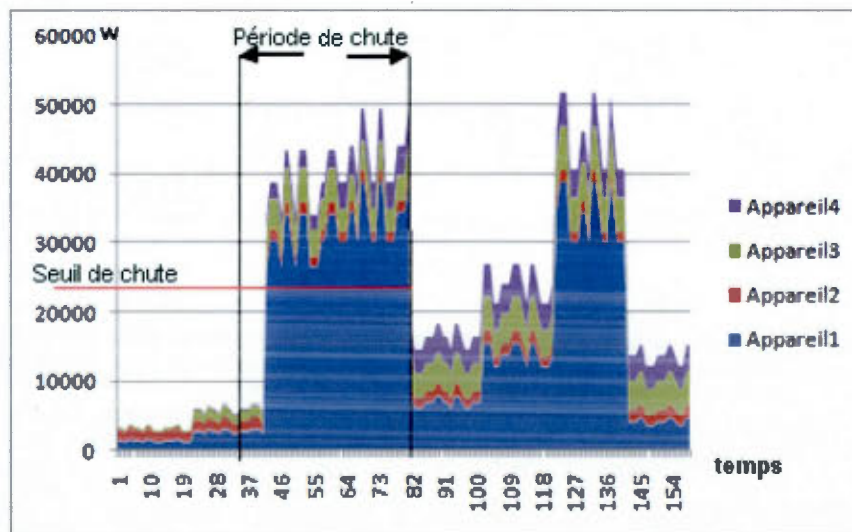


Figure 3.12 – La consommation des appareils ; le cas d'un réseau classique

D'après nos résultats, le réseau intelligent a consommé une quantité d'énergie égale à 2 930 729,325 W/jour. Dans un réseau classique et dans les mêmes circonstances, la consommation est de 3 799 222 W/jour. Le réseau classique ne s'est donc pas adapté à cette chute.

La différence entre les deux consommations est de 868 493,175 W/j, ce qui montre que dans le cas d'un réseau intelligent, le système diminue sa consommation pour éviter une perturbation de fonctionnement au niveau des appareils, alors qu'un réseau classique conserve sa consommation avec les perturbations de fonctionnement que cela cause.

3.5 – Contrôle de consommation par foyer et par zone

Les données qui représentent la consommation au niveau d'un foyer sont sauvegardées en temps réel dans le compteur intelligent, dans une base de données XML telle que décrite dans la section 3.3. Ces données peuvent être consultées par un service de *monitoring* ou par le prioritaire du compteur intelligent.

Un service de commande peut consulter ces données à travers le Web. Cela permet à un client de consulter sa consommation énergétique ou même de piloter à distance ses appareils électroménagers.

Dans ce cadre, nous avons implémenté une architecture en utilisant des services Web en nous basant sur le protocole SOAP. Nous avons créé des serveurs Web au niveau des foyers, ainsi que des serveurs au niveau des zones, selon l'architecture exposée dans la figure 3.12. Les serveurs dans les foyers sont accessibles à travers les serveurs Web en utilisant des adresses IP fixes assignées par l'opérateur du réseau électrique intelligent. Ces adresses sont supposées être connues par le *Centre de monitoring*. Le *Service de commande* de ce centre envoie une série de commandes formées de paires constituées du type d'opération désirée (consultation des données de consommations, maintenance à distance, etc.) et de l'adresse IP du compteur du foyer en question.

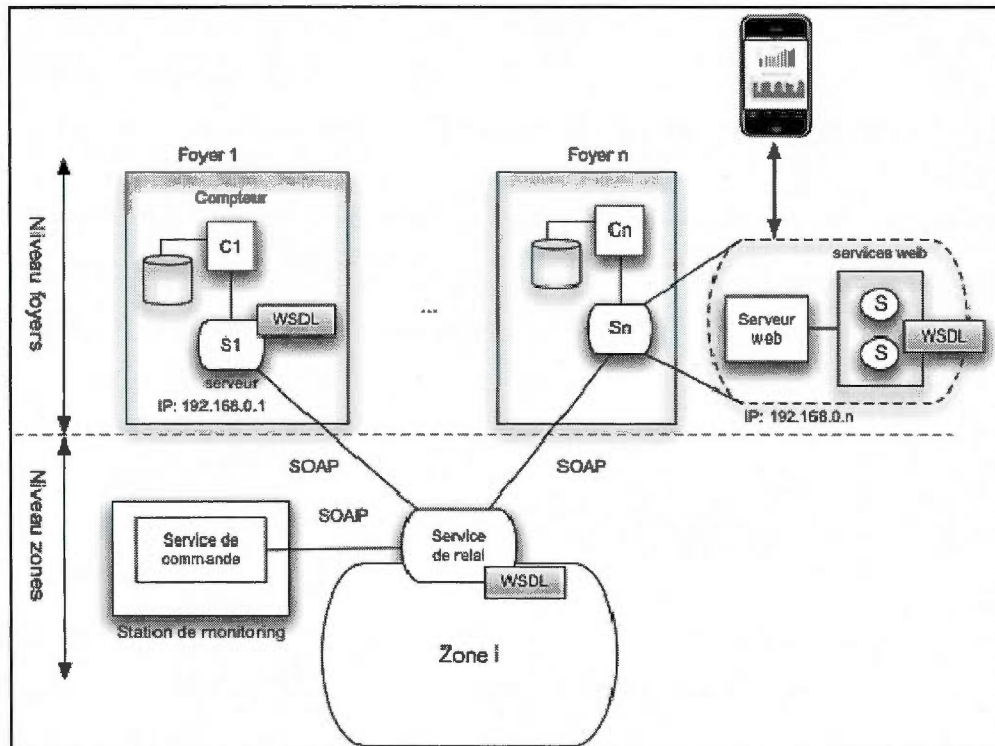


Figure 3.13 – Architecture de communication inter- zones

Un serveur sur un foyer contient des fonctions permettant l'accès aux bases de données. Chaque service dans cette architecture fournit sa description *WSDL* qu'il rend publique pour les autres services afin que ceux-ci puissent y accéder en fournissant les données nécessaires à cet accès.

Un module implémenté sur le serveur Web (notamment en *PHP*) accède aux services Web dans un foyer pour effectuer certaines tâches comme l'accès à la base XML des données du compteur local ou celles d'un compteur distant. Également, dans le centre de *monitoring*, le service de commande utilise les services Web publiés dans un *Service de relais* de la zone grâce à leurs descriptions *WSDL*. Ce relais peut faire appel aux services Web déclarés dans les serveurs attachés aux compteurs. Ces derniers publient également leurs services sous forme de fichiers *WSDL*.

Notons que dans le cadre de développement que nous avons utilisé (c'est-à-dire *Visual Studio* sous *.Net*), les descriptions *WSDL* ne sont pas publiées dans un registre séparé comme (*Universal Description, Discovery and Integration*) *UDDI*, mais dans un fichier qui se trouve sur un serveur Web (*Localhost* dans notre cas) identifié par son (*uniform resource identifier*) *URI*. Voici un exemple de l'utilisation d'un tel fichier :

```
< h1 >
< ?php
try{
    $client = new
        SoapClient( "http : //localhost : 8731/WcfServiceLibrary/Service/?wsdl" );
    $webService = $client-> GetData();
    $wsResult = $webService-> GetDataResult;

} catch (Exception $e) {
    print 'Caught exception :'. $e-> getMessage(). "\n";
}
?>
< /h1 >
```

Dans le serveur d'un compteur intelligent (c'est-à-dire dans un foyer), il existe plusieurs fonctions publiées sous forme de services telles que :

- *GetConsommationValueRealTime()* qui lit la consommation énergétique en temps réel pendant un intervalle Δt pour un appareil d'identité *AppID* dans un foyer :

```
GetConsumptionValueRealTime (int AppID)
{
    return Charge[AppID];
}
```

- *GetConsommationValue()* qui donne la consommation globale correspondant à la somme des consommations d'un foyer pendant un certaine durée :

```
double GetConsumptionValue(int AppID)
{
    return ChargeGI[AppID];
}
```

Grâce au module *System.ServiceModel* de la bibliothèque .NET, les fonctions publiées dans un fichier WSDL sont utilisées par le *Service de commande* pour obtenir des données afin qu'elles puissent être traitées et représentées sous forme de graphes en temps réel sur un poste client.

Le *Service de commande* peut ainsi lire la consommation globale du foyer en temps réel avec les détails de consommation pour chaque appareil et les afficher comme l'illustre la figure 3.14.

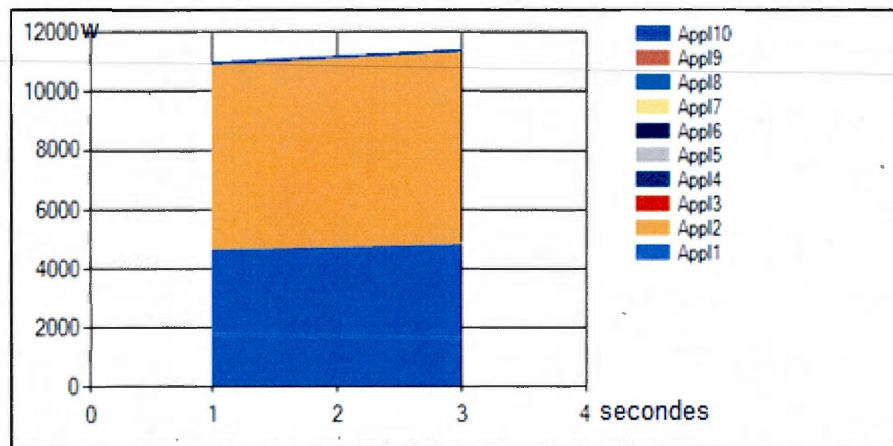


Figure 3.14 – La consommation énergétique par appareil dans un foyer

Dans cet exemple, le *Service de commande* a sélectionné un foyer en particulier (désigné par son adresse) afin de demander la consommation de l'énergie globale avec la désignation de chaque appareil, chacun étant représenté par une couleur différente.

Dans l'exemple de la figure 3.15, on procède de la même façon pour consulter la consommation d'une zone particulière. Ce résultat représente la somme des consommations

des foyers qui se trouvent dans cette zone. Chaque foyer de la zone est représenté par une couleur différente.

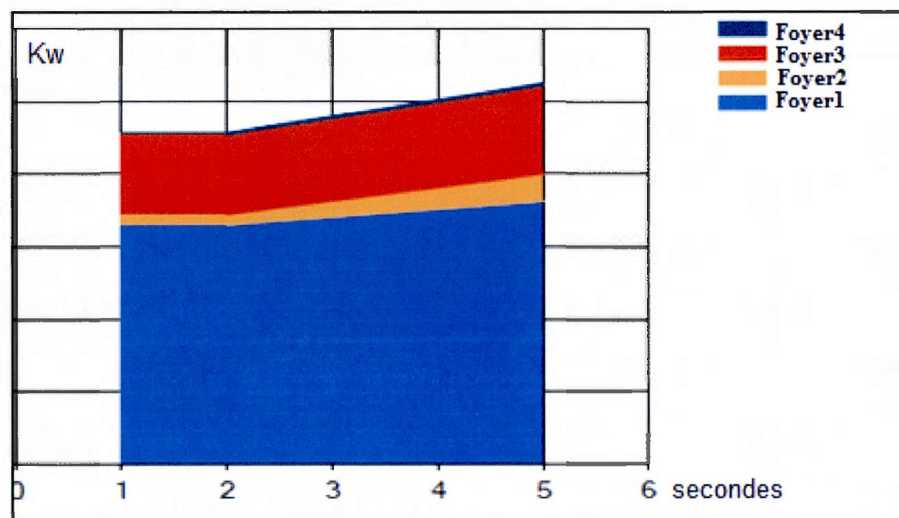


Figure 3.15 – La consommation au niveau d'une zone

3.6 – Contrôle à distance par appareil mobile

Nous avons proposé une architecture qui permet de consulter la consommation d'un foyer et de commander les différents appareils qui s'y trouvent. Ces deux activités ont été implémentées en tant que service Web.

Aux services qui existent déjà et que nous avons décrits, nous avons ajouté la tâche qui nous permet de commander un appareil à distance à l'aide d'un appareil mobile. Grâce à un mini-service installé dans le compteur intelligent et une interface réseau locale, on peut communiquer avec les appareils installés dans le foyer. Dans notre cas simulé, nous avons utilisé une liaison filaire sur le port *COM*. Avec un simple accès par mobile à la page de consultation, on peut lire la consommation en temps réel de chaque appareil dans le foyer simulé ainsi que la consommation globale en temps réel (Figure 3.16).



Figure 3.16 – La consommation d'énergie en temps réel dans un foyer

Le propriétaire du logement peut aussi commander ces appareils intelligents à distance, par exemple pour ajuster la température du foyer, éteindre un appareil, etc. (Figure 3.17).

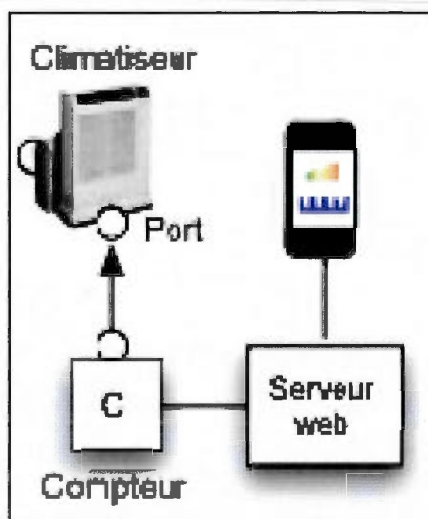
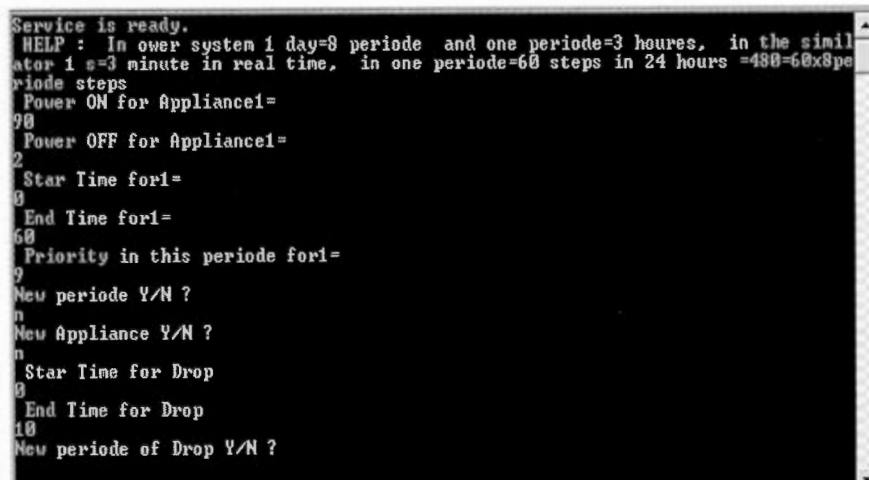


Figure 3.17 – La communication entre le mini-serveur et l'appareil

Pour relier ce système au monde externe (par exemple, un mobile), on connecte le serveur Web dans le foyer. Puisque ce serveur possède une adresse IP fixe, l'accès aux fonctions de

commande et de consultation sera possible. Un usager mobile accède à ce site par un code usager et un mot de passe.

Chaque appareil dans le foyer est représenté par une application console en C# qui utilise aussi cette technologie pour partager ses données avec l'application serveur. La configuration de chaque appareil prend en considération les données contenues dans le fichier XML de description des appareils comme l'état *POWER ON* ou *POWER OFF*, le taux d'utilisation de cet appareil durant chaque période de la journée et les périodes des chutes de tension (Figure 3.18).



```

Service is ready.
HELP : In over system 1 day=8 periode and one periode=3 heures, in the simil
ator 1 s=3 minute in real time, in one periode=60 steps in 24 hours =480=60x8pe
riode steps
Power ON for Appliance1=
98
Power OFF for Appliance1=
2
Star Time for1=
0
End Time for1=
60
Priority in this periode for1=
0
New periode Y/N ?
n
New Appliance Y/N ?
n
Star Time for Drop
0
End Time for Drop
10
New periode of Drop Y/N ?

```

Figure 3.18 – Simulateur d'un appareil intelligent

CHAPITRE IV

MÉCANISME DE NÉGOCIATION DE L'OFFRE ET DE LA DEMANDE D'ÉNERGIE

4.1 – Introduction

Dans ce chapitre, nous proposons un protocole d'échange pour la négociation de l'offre et de la demande d'énergie dans le réseau intelligent. Ce protocole est asynchrone et basé sur l'utilisation des agents mobiles. Il peut être décrit comme suit : un site (c'est-à-dire un foyer) cherche des fournisseurs d'énergie dans le réseau et choisit ceux qui offrent ce service (de vente d'énergie) à moindre coût et qui satisfont son comportement (ou *pattern*) de consommation. Il cherchera les fournisseurs sur le réseau, un par un, en procédant ainsi :

- (1) Pour un *pattern* de consommation donné, il lance une requête auprès de son agent mobile. Celui-ci traverse le réseau à la recherche de fournisseurs. Chacun de ces fournisseurs est représenté par son compteur intelligent.
- (2) Un foyer qui lance une requête est modélisé par un processus qui s'exécute sur son environnement, qui peut être un compteur intelligent dans le foyer, ou un routeur qui est connecté à la région où se trouve ce foyer.
- (3) Ce processus de négociation retournera le résultat au site du foyer demandeur. Ces données sont ensuite affichées sur l'écran avec les noms et les adresses des foyers qui proposent ce service au prix le plus bas, et qui satisfont le *pattern* du requérant. Le requérant peut alors choisir un fournisseur, annoncer l'acceptation de son offre et lui proposer un contrat d'achat.

Précisons que nous ne proposons pas de procédure de négociation en tant que tel. Ce sujet dépasse le cadre de ce travail. Il peut faire appel à des concepts de négociation que l'on trouve sur le marché des échanges économiques. Nous proposons simplement un mécanisme d'échanges et une infrastructure qu'une telle procédure peut éventuellement utiliser.

Suivant l'architecture du réseau que nous avons proposée, l'agent mobile utilise les compteurs intelligents et les routeurs pour suivre son itinéraire (Figure 4.1).

Par la suite, nous évaluerons la performance de notre solution dans un réseau sans fil, en considérant les mécanismes d'arrivée et de service de ces agents à des files d'attente. Cette étude nous permet d'évaluer la vitesse à laquelle les agents mobiles sont servis.

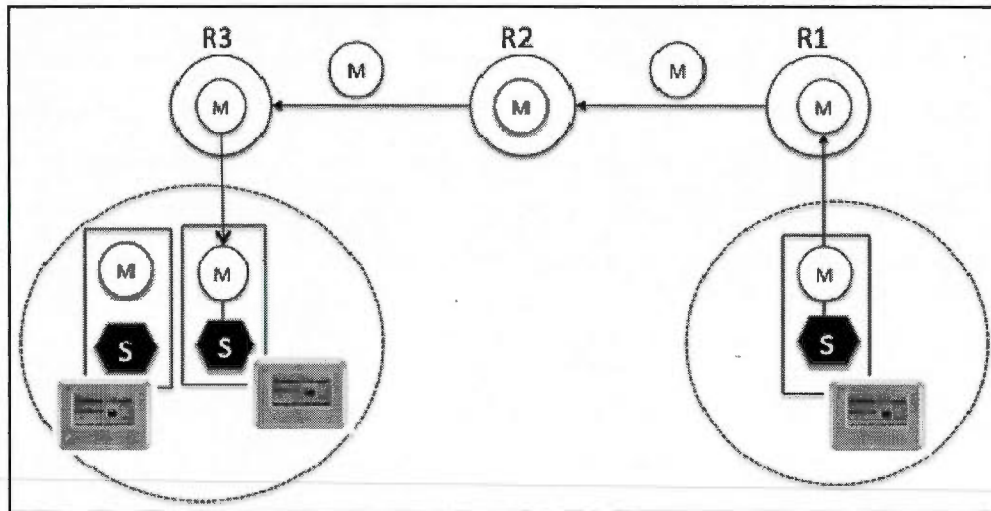


Figure 4.1– Itinéraire d'un agent mobile dans le réseau

Nous utilisons deux types d'agents : stationnaire et mobile (*S* et *M* dans la figure 4.1).

S représente un compteur intelligent et effectue des tâches comme :

- L'activation des agents mobiles et la spécification de leurs itinéraires.
- La mise en disponibilité des services offerts par un foyer et leur soumission à l'agent mobile. Un exemple d'un tel service est de fournir les informations sur l'énergie en surplus de ce que le foyer peut offrir aux autres foyers.
- La réception des informations sur les offres d'énergie que les fournisseurs déposent et qui sont portées par les agents mobiles.
- L'ordonnancement des appareils électroménagers dans le foyer. Cet aspect est décrit dans le chapitre 2.

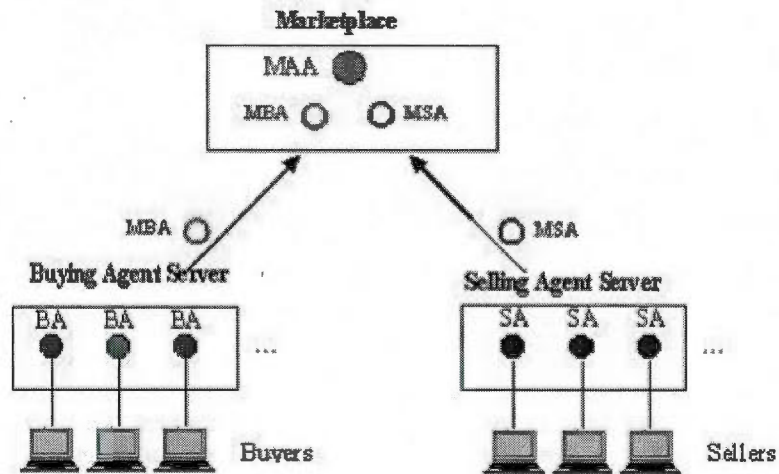
Un agent mobile peut passer d'un foyer à un autre ou d'une zone à une autre à travers les compteurs intelligents et les routeurs. Un agent mobile M , dans sa zone d'origine, sera envoyé par l'agent stationnaire S du compteur intelligent. Le premier site visité est le routeur R_1 qui représente la zone où se trouve le compteur initial S . Ce dernier cherche s'il y a des fournisseurs d'énergie dans sa zone. S'il y en a, le routeur R_1 renvoie une copie d'agent vers les routeurs voisins et aux foyers fournisseurs de sa propre zone ; sinon il envoie une copie d'agent vers les routeurs voisins.

La même procédure s'exécute ainsi par des envois en parallèle d'agents mobiles.

4.2 – Travaux connexes

Avant de décrire notre proposition par rapport à cet aspect, nous allons décrire présenter travaux faits par d'autres chercheurs et les comparer au notre.

Dans le modèle *e-Marketplace* proposé par P. Dasgupta [47], un client envoie une requête de type agent mobile basé sur une plateforme Java Aglet [48] à un fournisseur à travers un serveur commun nommé «Marché», afin de négocier sur un ou plusieurs produits publiés sur cette plateforme virtuelle. Ce modèle est basé sur trois parties. Un serveur d'agents mobiles associé au client, un serveur associé aux agents mobiles de fournisseur, et un serveur de marché qui reçoit les agents clients et les agents fournisseurs afin d'exécuter les tâches de négociations d'achats et de ventes à ce niveau (Figure 4.2).

Figure 4.2 – Le modèle *e-Marketplace* [48]

Dans ce modèle, les requêtes de négociations s'exécutent sur un serveur centralisé (Marché). Notre proposition de négociation résoud ce problème car au lieu que ce soit les requêtes de négociation qui s'exécutent dans un serveur centralisé, elles s'exécutent au niveau des serveurs fournisseurs.

Un autre modèle est proposé par des chercheurs de l'Université Chaoyang [49] (Figure 4.3).

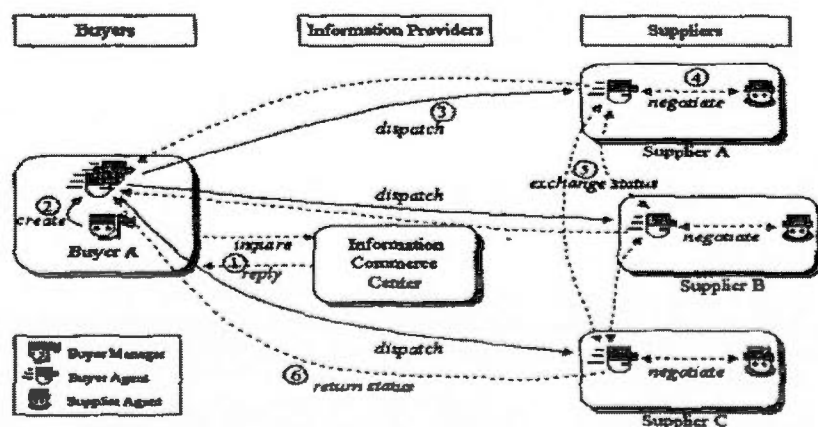


Figure 4.3 – Coopérative multi-agent de négociation [49]

Cette proposition résout le problème de la centralisation, et propose une autre solution : Un client envoie plusieurs agents mobiles vers des serveurs fournisseurs sans intermédiaires, et les opérations de négociations s'exécutent aux niveaux de ces derniers. Comparé avec notre proposition, dans notre cas un client envoie un agent vers plusieurs fournisseurs, et des copies de cet agent seraient créés aux niveaux des routeurs de zones. Ces derniers (agents) ne visitent que les zones qui contiennent des sites fournisseurs.

Un autre travail est proposé par des chercheurs de l'université japonaise Hosei [50]. Il représente une amélioration du modèle « MineAgeHops » [50] (Figure 4.4).

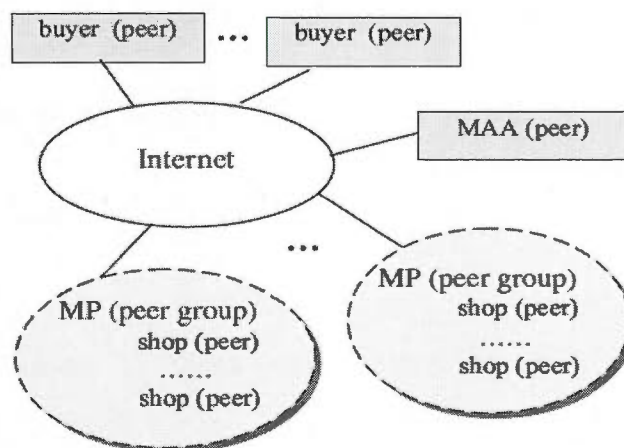


Figure 4.4 – Schéma décentralisé (MineAgeHops) [50]

Ce modèle est basé sur le MISS (Marketplace Information Service Server), le MAA (Marketplace Assessment Authority), et les MPs qui représentent les vendeurs et les acheteurs sur le modèle. Un site client (acheteur) peut communiquer avec les autres sites fournisseurs (vendeurs) passant par un serveur commun. Ce serveur surveille les opérations des recherches d'un ou des fournisseurs pour l'acheteur. Après avoir obtenu les informations sur les vendeurs, l'agent acheteur accède aux sites vendeurs afin de négocier sur un ou plusieurs produits publiés sur le réseau. À la fin des opérations, l'agent apporte les informations au site acheteur et les affiche sur une interface visuelle.

La première chose à discuter au sujet de ce modèle est le rôle du serveur cherchant les vendeurs pour un agent acheteur. Un serveur centralisé qui gère plusieurs opérations

simultanément (vendeurs/acheteurs) pose toujours les problèmes de centralisation et de congestions de données dans le réseau.

La deuxième chose est la taille des données apportées par un agent mobile. La taille des données apportées par un seul agent vers le site acheteur représente toujours un grand défi dans le domaine des agents mobiles.

Dans notre proposition, un site acheteur envoie un agent mobile vers des sites vendeurs et non plusieurs agents. Il reçoit ainsi plusieurs agents qui apportent les informations. Au lieu que cela soit tous les agents acheteurs et vendeurs qui accèdent au même serveur, notre proposition remplace ce serveur par des routeurs installés au niveau des zones. Chaque routeur d'une zone procède par deux tables de routages, une représentant les sites vendeurs dans sa zone, l'autre représentant les routeurs des zones voisines.

Dans MAgNET [51] (Figure 4.5), un acheteur possède une liste des vendeurs et de leurs produits publiés. Au début, l'acheteur envoie un agent mobile qui porte dans son corps une liste des chemins vers les vendeurs choisis, ainsi que les informations des produits voulus. Cet agent visite site par site, et exécute la procédure de négociation au niveau de chaque site vendeur et empile le résultat de négociation dans son corps. Suivant la liste de routage, il visite les autres vendeurs et exécute le même scénario. Finalement, il retourne vers son site initial (site acheteur) par des données ramassées, et le client donne finalement sa décision.

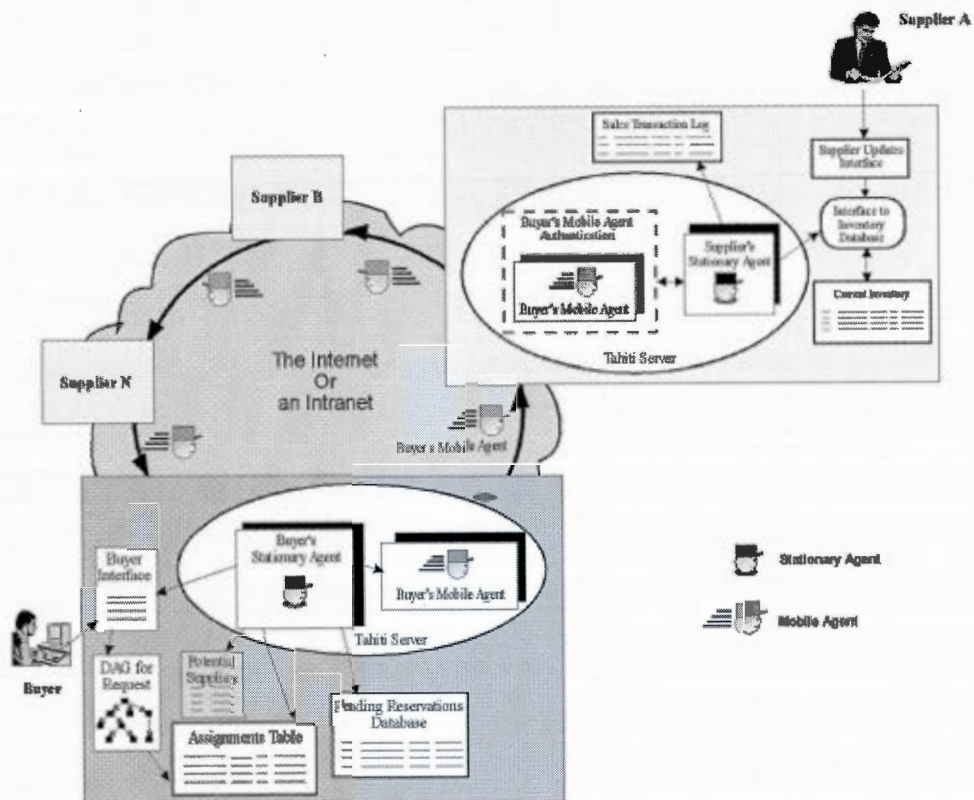


Figure 4.5 – Architecture MAgNET [51]

Comparé à notre modèle, dans cette proposition un agent envoyé par un site acheteur empile une liste des chemins vers les vendeurs dans le site initial (site d'acheteur). Par contre, dans notre proposition l'agent n'empile que l'adresse du routeur de sa zone, ensuite il empile les autres chemins aux niveaux des routeurs visités. La deuxième chose, dans le protocole de modèle étudié le même agent apporte les données ramassées des fournisseurs, ce qui cause l'augmentation de sa taille. Par contre, dans notre proposition un agent sera envoyé pour négocier et plusieurs apportent les informations vers le site initial.

Dans la littérature, nous n'avons pas trouvé un modèle actuel qui ressemble à notre proposition. Cette proposition est basée sur cinq notions :

1. Un agent est envoyé vers les vendeurs et plusieurs agents apportent les informations.

2. L'agent dans son départ empile dans son vecteur de routage que l'adresse du routeur de sa zone.
3. L'agent empile les chemins vers les vendeurs dans son vecteur de routage aux niveaux des routeurs visités.
4. Dès qu'un agent trouve des vendeurs dans la zone visité, une copie de ce dernier sera envoyée vers les routeurs voisins, et l'autre copie sera envoyée vers les sites vendeurs de cette zone, avec une condition qu'il retourne vers son site initial dis que il sort de dernier fournisseur de cette zone.
5. Finalement, tous les agents qui apportent les informations retournent vers le site initial, et les informations serrent affichés sur un écran, et l'acheteur prend les décisions. Les agents qui ne trouvent pas des vendeurs ont une durée de vie.

4.3 – Le *middleware*

L'agent mobile dans notre proposition s'exécute dans les routeurs et les compteurs intelligents. Cette exécution est prise en compte par un *middleware* installé dans chacun de ces nœuds. Ce *middleware* permet à plusieurs processus *agents* de s'exécuter sur les machines visitées et d'interagir à travers un réseau [35].

Chaque compteur intelligent est relié à un terminal qui comporte ce *middleware*. Ce terminal est implémenté dans le mini-serveur proposé dans notre architecture (voir Chapitre 2).

Lorsque le compteur reçoit l'agent mobile, il le renvoie au terminal sur un lien *ZigBee* (ou une connexion RJ45) afin qu'il s'exécute dans le *middleware*. Le compteur se charge de la transmission et de la réception des agents. Le terminal exécute l'agent et le renvoie au compteur afin qu'il le retransmette sur le réseau. Ce mécanisme est représenté par la figure 4.6.

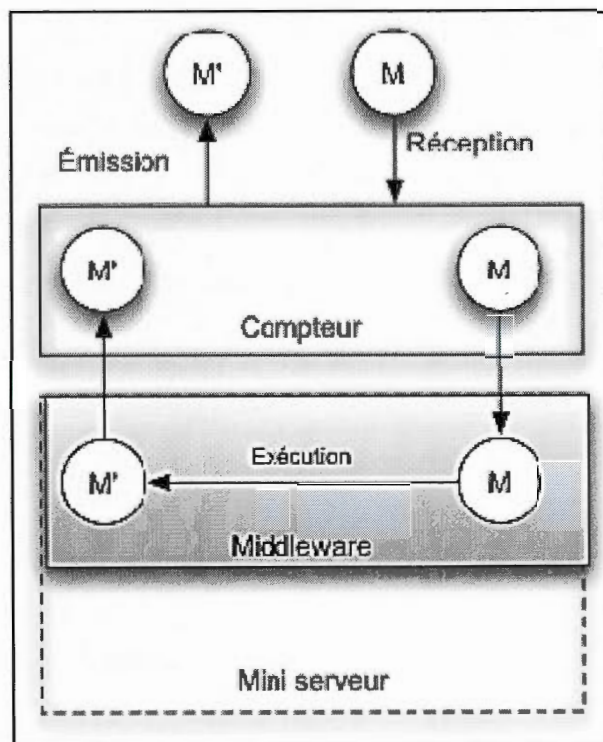


Figure 4.6 – La relation entre le compteur intelligent et le *middleware*

Plusieurs fonctions sont offertes par le *middleware* :

- Une fonction qui écoute sur un port de communication l'arrivée d'un agent mobile.
- Une fonction qui donne accès à la base de données du compteur intelligent.
- Une fonction de décryptage utilisant une clé de chiffrement contenue dans l'agent mobile pour décrypter le code exécutable (celui-ci est un texte représentant des instructions écrites en un langage de programmation connu par le *middleware*).
- Une fonction qui exécute l'agent dans le *middleware* après le décryptage.
- Une fonction qui duplique l'agent en plusieurs copies pour les envoyer aux autres routeurs ou compteurs.
- Une fonction qui renvoie l'agent vers le compteur intelligent afin qu'il le retransmette sur le réseau.

Le *middleware* utilise deux files d'attente pour stocker les agents mobiles : une file pour les agents en état de recherche de services (*Buffer1*) et une autre pour les agents en état de retour vers leurs foyers d'origine (*Buffer2*). Les agents dans *Buffer1* seront servis afin qu'ils soient envoyés vers d'autres routeurs sur le réseau à la recherche de fournisseurs d'énergie. Les agents dans *Buffer2* sont porteurs d'informations vers des foyers qui sont prêts à offrir de l'énergie. Chacun de ces agents se dirige vers le foyer du requérant qui a amorcé cette demande d'énergie (Figure 4.7).

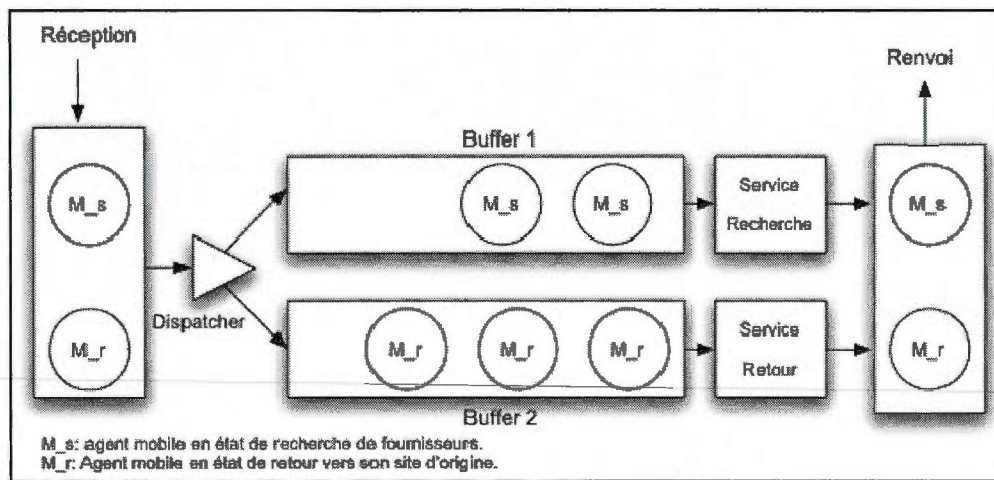


Figure 4.7 – Traitement des agents dans un routeur

4.4 – Structure d'un agent mobile

Un agent mobile est structuré sous forme de paquet qui comporte les principaux champs suivants :

- (1) L'entête qui contient le champ *id* (identificateur de l'agent), le type (agent de négociation dans notre cas, agent de contrats, agent de prélèvements) et la *taille* (taille du paquet de l'agent).
- (2) Le code exécutable qui sera pris en charge par les nœuds visités.

- (3) Une clé de chiffrement utilisée pour décrypter le code exécutable dans les terminaux visités. Cette clé est utilisée par le foyer initial afin de crypter le code de l'agent et de le protéger contre des modifications malveillantes.
- (4) L'adresse qui identifie le site initial qui a créé et envoyé l'agent.
- (5) Un vecteur de routage qui contient les adresses des foyers et des routeurs à visiter.
- (6) Un tableau de 55 champs qui est un vecteur de l'historique du *pattern* de consommation (Figure 4.8).
- (7) Les données collectées qui seront reçues par le site initial. Une donnée représente l'historique de service publié par un fournisseur durant une année, le prix d'unité de l'énergie offerte par ce fournisseur et le chemin vers le site de ce fournisseur. Grâce à ces données, le client peut choisir le fournisseur qui lui convient et émettre un contrat d'achat avec lui.

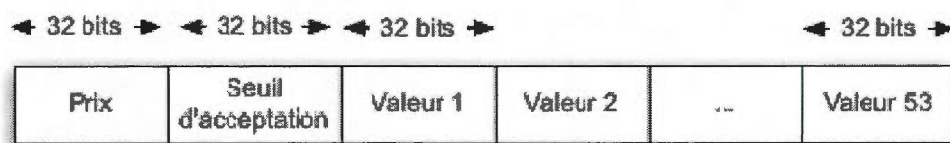


Figure 4.8 – Vecteur d'historique d'énergie utilisée par le foyer client

L'agent mobile proposé prend une taille de 27528 octets, incluant les données transportées à la suite d'une opération de négociation. Un schéma détaillé d'un paquet d'un agent est représenté par la figure 4.9.

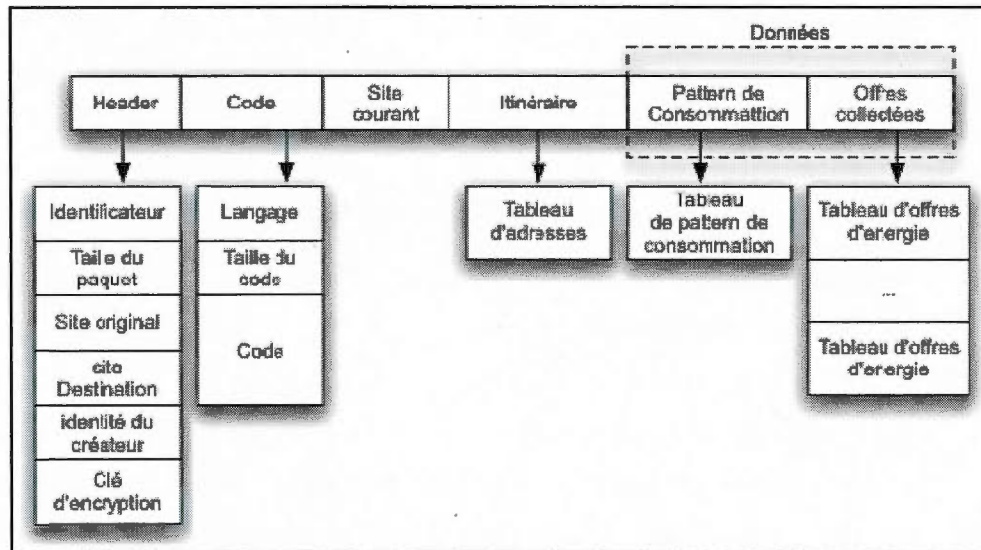


Figure 4.9 – Structure de paquet d'un agent mobile

Lors de son parcours, l'agent ramasse les données fournies par les foyers qui fournissent de l'énergie. Ces données transportées par un agent peuvent avoir une grande taille. Pour cette raison, nous avons proposé un scénario qui limite à 536 octets la taille des données ramassées pour chaque foyer. L'agent mobile peut regrouper les données ramassées de 20 foyers au maximum. Puisque la taille des données ramassées par l'agent est limitée à 20×536 bytes, un agent doit visiter une seule zone. Le nombre maximum de foyers fournisseurs qui seront visités est limité à 20. Grâce à la procédure de duplication d'agent au niveau des routeurs, le même corps d'agent envoyé par le foyer initial sera envoyé aux autres routeurs voisins qui représentent les autres zones.

L'agent porte un *Vecteur de consommation* qui représente l'historique d'utilisation d'énergie (ou *pattern* de consommation) du foyer du requérant durant une année. Ce vecteur est structuré selon la figure 4.8. Le champ *Prix* indique le prix maximal que le requérant est prêt à accepter. Le champ *Seuil d'acceptation* indique le pourcentage de service accepté par rapport à son *pattern* de consommation d'énergie par le foyer requérant. Il représente le pourcentage du service fourni par rapport à son besoin. Par exemple, on peut y indiquer que le foyer client accepte un service qui respecte 90 % de son besoin. Suit une série de 53

valeurs, chacune représentant la quantité d'énergie consommée par le foyer durant une semaine.

Ce vecteur sera utilisé par la fonction de négociation pour permettre au foyer fournisseur d'énergie d'accepter ou de refuser l'offre. Ce vecteur sera comparé au vecteur qui représente le service offert et qui se trouve dans la base de données du foyer fournisseur. Un foyer fournisseur peut posséder un générateur d'énergie (renouvelable ou non). Ce foyer consomme une partie de cette énergie et peut vendre celle qu'il a en surplus. L'énergie produite par le générateur sera stockée dans un support quelconque (par exemple une batterie rechargeable). L'énergie qui sort de la batterie sera transformée en courant alternatif afin qu'elle puisse être utilisée par les appareils électroménagers du foyer. Avant que cette énergie ne soit utilisée, elle passe par un comparateur qui compare l'énergie fournie par le générateur et l'énergie consommée par le foyer. La différence constitue le surplus que le foyer peut rendre disponible aux autres consommateurs *via* la procédure de négociation (Figure 4.10).

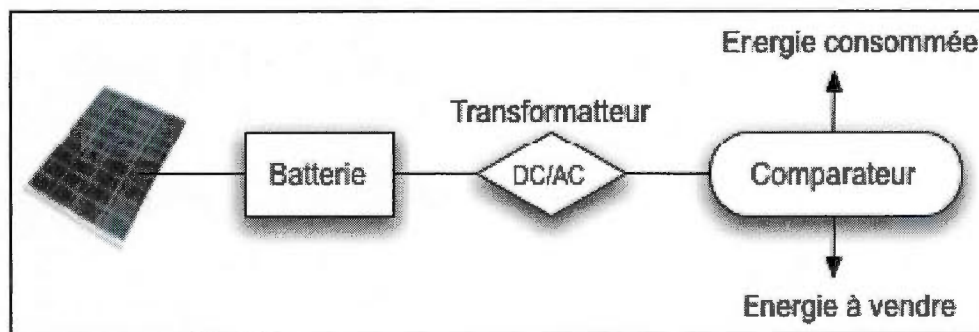


Figure 4.10 – Mécanisme de consommation et de vente d'énergie

Le fichier de type XML du fournisseur, qui représente l'historique du service de vente au niveau de la base de données du compteur intelligent, sera créé en fonction de l'énergie disponible en surplus. Ce fichier contient la quantité d'énergie produite localement, les pourcentages d'offre pour chaque semaine de l'année par rapport à la totalité d'énergie produite localement (mise en vente) ainsi que le prix d'unité. Ces données sont représentées comme suit (Figure 4.11).

| | | | | | |
|---------------------------|---------------------------|-------------------|-----|----------------------------|--------------------|
| Qté d'énergie produite | % disponible Semaine 1 | Prix Semaine 1 | ... | % disponible Semaine 53 | Prix Semaine 53 |
|---------------------------|---------------------------|-------------------|-----|----------------------------|--------------------|

Figure 4.11 – Vecteur d'historique d'un service d'énergie produite localement

L'agent appelle la fonction *match* () qui utilise les deux vecteurs de négociation : le vecteur de consommation apporté par l'agent et le vecteur de service au niveau du foyer fournisseur. Cette fonction retourne *true* ou *false*, selon que le service est accepté ou non par le requérant. Si le service est accepté, les données offertes par le fournisseur sont embarquées sur l'agent mobile, sinon l'agent mobile quitte ce foyer pour aller aux autres foyers fournisseurs suivant son vecteur d'adressage. Pour ce faire, l'agent reconstruit le vecteur de routage en dépilant son dernier élément. Pour faire le tour des foyers qui fournissent l'énergie dans une zone, l'agent utilise le vecteur d'adresses tel qu'illustré dans la figure 4.12. Cette figure montre une zone contrôlée par le routeur *R1* et qui comprend 4 foyers (*F1*, *F2*, *F3* et *F4*). Seuls *F1*, *F2* et *F3* offrent de l'énergie.

| | | | | | |
|-------------------------|------------------|------------------|---------------|---------------|---------------|
| Adresse du requérant | Adresse du R2 | Adresse du R1 | Adresse F1 | Adresse F2 | Adresse F3 |
|-------------------------|------------------|------------------|---------------|---------------|---------------|

Figure 4.12 – Vecteur de routage dans un agent mobile

Une fois arrivé à son site original, l'agent dépose les données rassemblées dans une base de données locale XML. Ces données contiennent l'itinéraire vers un site fournisseur, la quantité d'énergie fournie, le pourcentage d'offre d'énergie dans chaque semaine de l'année par rapport à l'énergie fournie, et le prix unitaire de l'énergie pour chaque semaine. Dans le cas où le client sur le site original sélectionne et accepte un fournisseur particulier, il informe celui-ci de son acceptation et informe les autres fournisseurs de son désistement. Par la suite, il pourra signer un contrat d'achat avec lui.

4.5 – Le protocole de négociation

Le protocole de négociation est basé sur le parcours illustré par la figure 3.17. Initialement, l'agent se trouve dans son site initial (celui du foyer requérant). Il passe ensuite à travers les étapes suivantes :

- (1) Il ajoute l'adresse du site initial et l'adresse du routeur de sa zone dans son paquet.
- (2) L'agent est envoyé vers ce routeur de zone (l'état d'agent mobile à cette étape est représenté par la figure 4.13).

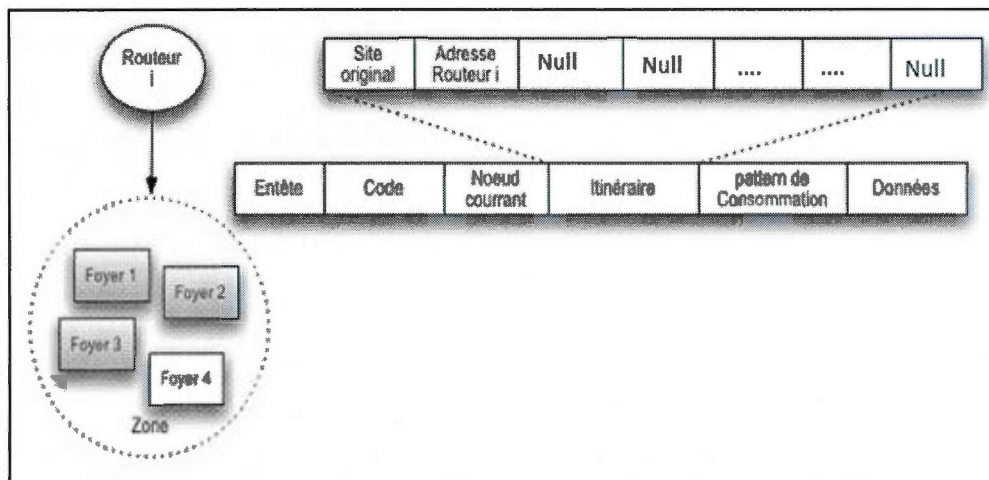


Figure 4.13 – L'agent mobile dans l'état initial

- (3) Le routeur reçoit l'agent.
- (4) Le code de l'agent s'exécute dans le *middleware* du routeur. Ce dernier vérifie la valeur de la variable *Retour* dans le corps d'agent. Cette variable booléenne indique si un agent est ou n'est pas dans un état de retour vers son site initial.
 - (i) Il duplique le corps d'agent en créant l'agent *c_Agent*.
 - (ii) Il ajoute au vecteur de routage de *c_Agent* l'adresse du routeur courant ainsi que les adresses des foyers qui fournissent de l'énergie.
 - (iii) Il empile l'adresse du routeur courant dans le vecteur de routage de *c_Agent*.

- (iv) Il envoie l'agent au premier site fournisseur qui figure dans ce vecteur de routage.
- (v) Il envoie *c_Agent* aux routeurs voisins du routeur courant. (Dans ce cas, l'agent se branche à l'étape 3.)
- (vi) Dans le premier foyer fournisseur visité, le code exécutable d'agent change la valeur de la variable *Retour* à *true*. (Cette valeur demeure égale à *true* durant chacune des prochaines étapes.)
- (vii) L'agent dépile l'entête de son vecteur de routage.
- (viii) L'agent lance la fonction de négociation. (Cette fonction retourne une décision : service accepté ou service non accepté.)
 - (1) Si le service est accepté, le code exécutable utilise les fonctions du *middleware*, lequel lui permet d'accéder aux bases de données pour empiler les données du service publié ainsi que le vecteur de routage vers ce foyer fournisseur.
 - (2) Si le service n'est pas accepté par l'agent, il n'empile pas les données de service.
- (ix) L'agent sera envoyé vers le site ou le routeur qui se trouve à l'entête de liste du vecteur de routage.
- (x) Si l'agent est reçu par un autre site fournisseur, les étapes vii, viii et ix se répètent. L'agent, dans ce cas, prend la forme suivante (Figure 4.14).

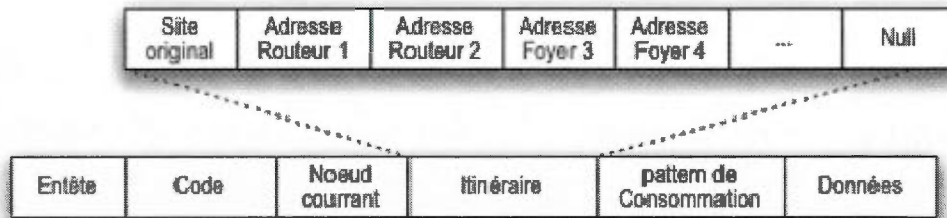


Figure 4.14 – Structure de routage de l'agent mobile

- (xi) Si l'agent est reçu par un routeur, on passe à l'étape 3.

(6) Dans le cas où la variable *Retour* est à *false* et s'il n'existe pas de sites fournisseurs dans la zone, le scénario suivant sera exécuté :

- (i) Le code d'agent empile l'adresse du routeur courant dans son vecteur de routage.
- (ii) L'agent sera envoyé par le routeur courant vers les routeurs voisins.
- (iii) L'algorithme reprend ensuite à l'étape 3.

Un exemple de ce protocole est donné par la figure 4.15.

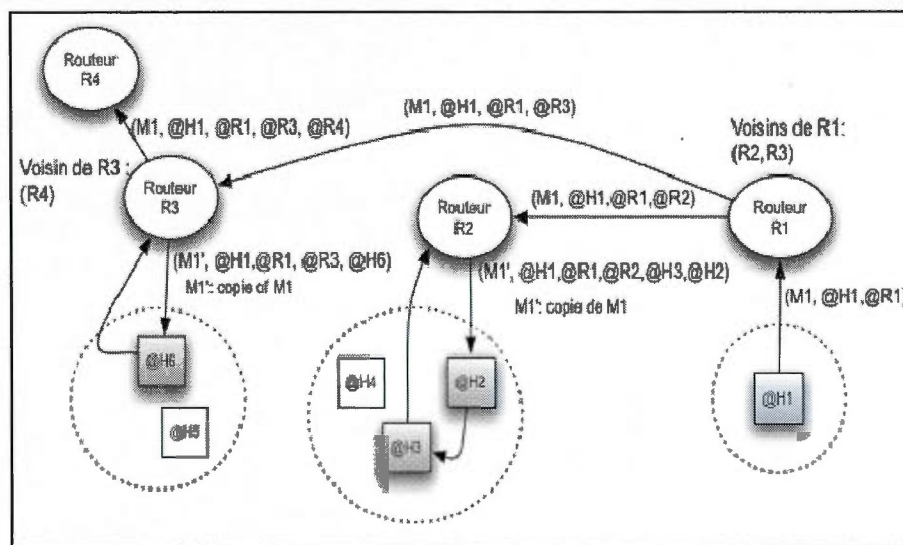


Figure 4.15 – Exemple de routage d'un agent mobile

4.6 – Implémentation

Dans notre implémentation, nous avons utilisé le langage C# sous *Visual Studio 2010*. Dans le *Framework* (version 4.0) de cet environnement, il existe une fonction qui permet d'exécuter le code d'un agent à partir d'un fichier texte, ou même à partir d'un flux de type *Chaîne de caractères* reçu à travers un port de communication. On utilise cette fonction pour créer un environnement exécutant ce flux de données reçu sur un port de communication qui représente l'agent mobile. Le flux est reçu sur le port via une interface *socket*.

Nous avons créé des applications qui représentent le comportement d'un compteur intelligent, d'un routeur et de l'environnement d'exécution des agents pour démarrer l'exécution, un foyer initial envoie un agent vers le routeur le plus proche (tel qu'illustré dans la figure 4.16) :

```
Input = Body.Text; // Corps d'agent mobile
server.Send(Encoding.ASCII.GetBytes(input));
```

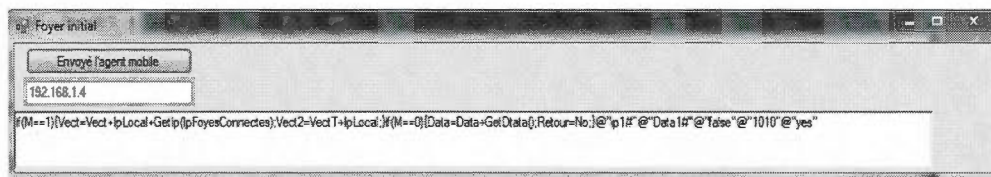


Figure 4.16 – Envoi d'un agent mobile à partir du site initial

Le routeur reçoit l'agent sur un port de communication comme suit :

```
data = new byte[1024];
recv = client.Receive(data);
```

Dans le *middleware*, le corps d'agent reçu sera analysé afin de lire son contenu :

```
Code = Encoding.ASCII.GetString(data, 0, recv);
string[] T = Code.Split('@');
if (T.Length > 1) {
    string CodeExec = T[0];
    string Vect = T[1];
    string Data = T[2];
    string Retour = T[3];
    string Signature = T[4];
    string Initial = T[5];
    string CodeSource = "CodeS= \"\" + CodeExec + \"\" ;";
```



```
}
```

Les données extraites seront envoyées vers les fonctions de négociation au niveau du middleware :

```
MidellWare(CodeSource, CodeExec, Vect,Data,Retour,Signature,Initial);
```

Le code exécutable (CodeExec) sera exécuté sur le *middleware* par un *API framework* :

```
CSharpCodeProvider provider = new CSharpCodeProvider(providerOptions);
```

```
CompilerParameters compilerParams = new CompilerParameters
```

```
...
```

```
CompilerResults results=
```

```
provider.CompileAssemblyFromSource(compilerParams, Agent[0]);
```

Par un clic sur le bouton *ReSend Mobile agent*, l'agent sera envoyé vers le foyer *Fournisseur 1* et aux autres routeurs voisins. Une indication de réception est alors affichée (Figure 4.17).

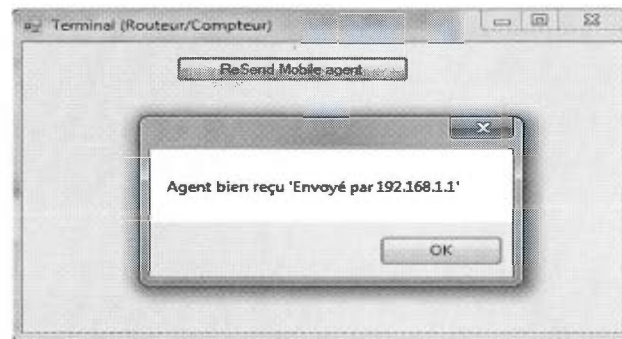


Figure 4.17 – Réception d'agent par un routeur

Le foyer initial reçoit finalement (par la même procédure d'envoi et de réception d'agent) toutes les données obtenues des foyers fournisseurs après négociations. Ces données représentent un service fourni durant quatre semaines (Figure 4.18).

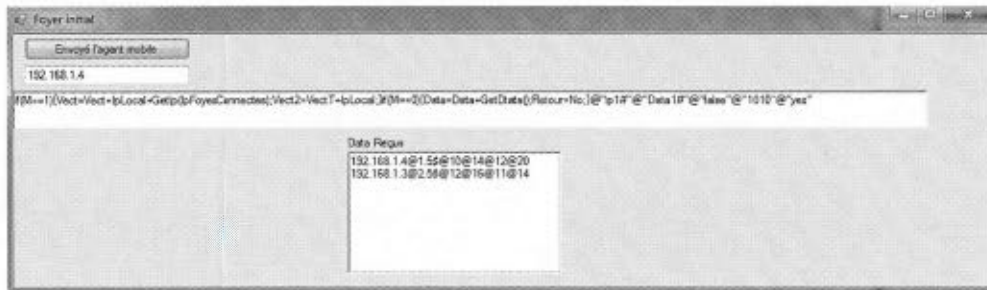


Figure 4.18 – Réception des données par le foyer initial

CHAPITRE V

ÉVALUATION DU PROTOCOLE D'ÉCHANGE DES AGENTS MOBILES

5.1 – Introduction

Dans ce chapitre nous décrivons notre implémentation d'un simulateur du protocole de négociation décrit dans le chapitre 3. Cette simulation a été réalisée avec l'outil *Java Modelling Tools* (ou JMT) [20].

Tel que décrit précédemment, chaque routeur possède deux files d'attente :

- une file qui regroupe les agents qui cherchent un fournisseur d'énergie dans l'état *Aller*; et
- une file qui regroupe les agents en état *Retour* vers le foyer initial et qui transporte des données.

Afin d'éviter les pertes de données au niveau des routeurs, les capacités des deux files devraient être en mesure de gérer le bon déroulement de l'algorithme. Un routeur parfait est un routeur où la perte de données au niveau des deux files est négligeable. Nous étudions ce comportement par simulation. Nous avons proposé un réseau de cinq routeurs. Le routeur initial qui reçoit des agents mobiles est caractérisé par un flux suivant une loi normale.

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Le taux des arrivées des agents vers le routeur 1 est représenté par $\lambda = 1$ et le taux de service des agents au niveau du routeur est représenté par $\mu = 2$. Les agents dans les deux files seront servis par une politique de type premier-arrivé-premier-servi (*First In First Out*, FIFO). Un routeur dans notre modèle supporte un maximum de 1000 agents en mémoire.

D'un autre côté, un compteur intelligent contient l'infrastructure pour exécuter les agents. Ces agents sont d'abord mis dans une file d'attente avant d'être exécutés dans le middleware de ce compteur. Chaque file d'attente a une capacité maximale de 20 agents.

Dans notre simulateur, nous avons 5 routeurs utilisés comme suit :

- *Routeur 1* reçoit des agents avec un flux qui suit une loi normale avec le taux d'arrivées $\lambda = 1$ et le taux de service $\mu = 2$. La zone associée à ce routeur ne contient pas de foyers fournisseurs d'énergie. Dans ce cas, il crée des copies de chaque agent reçu et les envoie aux autres routeurs qui se trouvent dans la table des routeurs voisins. Chacun de ces routeurs reçoit une copie d'agent. Le taux de service (exécution d'agent mobile) dans chaque routeur et chaque compteur intelligent suit également une loi normale.
- Dans la zone du *Routeur 2*, il existe 20 foyers fournisseurs ($F1, F2 \dots F20$). Dans la zone associée au *Routeur 3*, on a deux foyers fournisseurs $F21$ et $F22$. La zone associée au *Routeur 4* possède trois foyers fournisseurs $F23, F24$ et $F25$. La zone du *Routeur 5* possède cinq fournisseurs ($F26$ à $F30$).
- Le *Routeur 2* a comme voisin le *Routeur 5* : il lui envoie la copie d'agent M' représentée par l'agent M'' . Ce modèle est représenté par la figure 5.1.

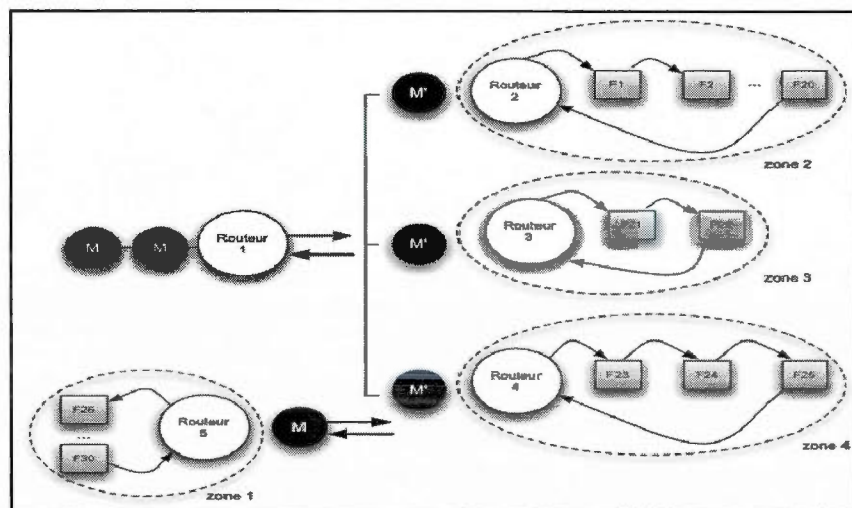


Figure 5.1 – Schéma du modèle simulé

Le modèle sur le simulateur (*Java Modelling Tools*) JMT est représenté dans la figure 5.2 :

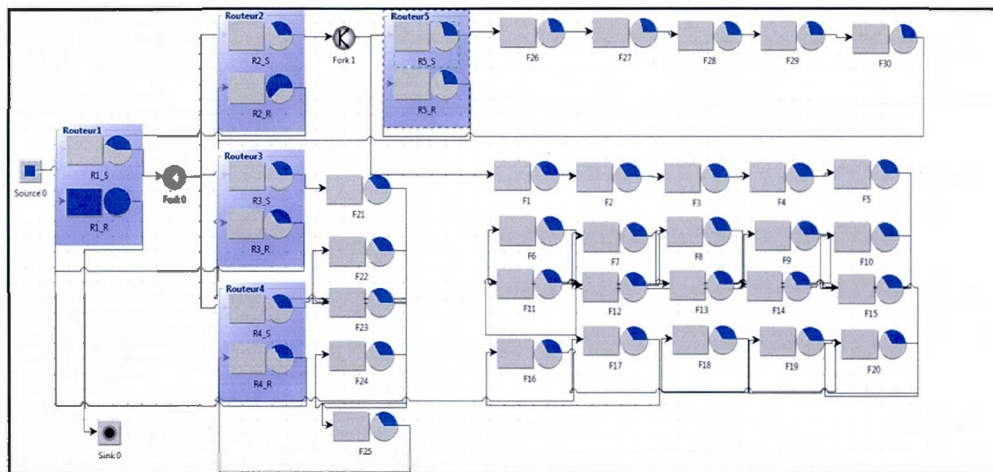


Figure 5.2 – La représentation de modèle simulé sur le simulateur

Nous avons fixé la taille des files d'attente pour chaque routeur et chaque compteur intelligent comme suit :

- La taille des files d'attente qui reçoivent des agents mobiles dans l'état *Recherche de fournisseurs* est fixée à 100.
- La taille des files d'attente qui reçoivent des agents mobiles dans l'état *Retour* est fixée à 1 000.
- La taille des files d'attente qui reçoivent des agents mobiles au niveau des compteurs intelligents est fixée à 20.

Dans cette configuration, la simulation du système s'est déroulée durant 20 secondes pour donner les résultats dans la section suivante.

5.2 – Résultats de simulations

5.2.1 – Au niveau du *Routeur 1*

Sur ce plan, nous avons obtenu les résultats observés au niveau des deux files d'attente. Au niveau de la première file, le résultat qui exprime le nombre d'agents mobiles dans la file d'attente par rapport au temps est donné dans la figure 5.3.

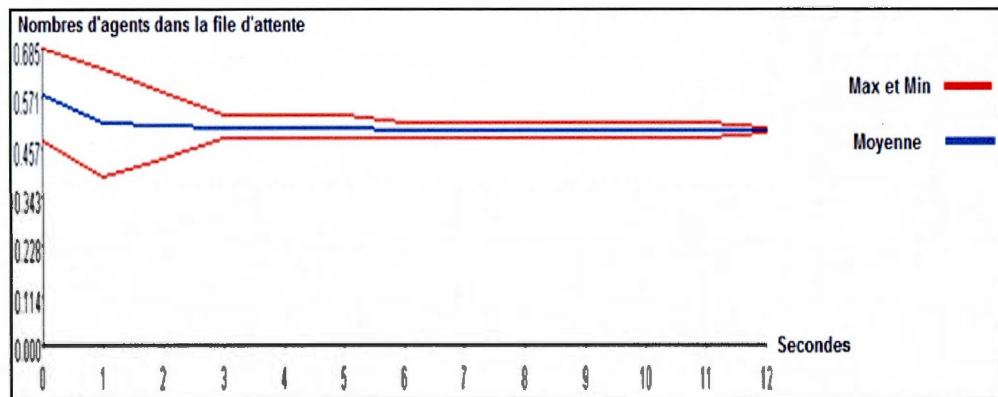


Figure 5.3 – Nombre d'agents sur la *File d'attente 1*
dans le *Routeur 1* en fonction du temps

La figure 5.3 montre que durant 12 secondes, le simulateur a estimé le nombre moyen d'agents qui attendent sur la file d'attente. Cette valeur tend vers 0,4958 : cela veut dire qu'il n'y a pas d'attente à ce niveau.

On peut en conclure que la taille de cette file peut prendre une valeur moindre que 100. Par exemple, elle peut prendre une valeur comprise entre 2 et 10, et minimiser la taille de mémoire dans le routeur réservé au traitement des agents mobiles dans cet état.

Les résultats obtenus dans la deuxième file d'attente (agents en état *Retour*) sont représentés par le graphe de la figure 5.4.

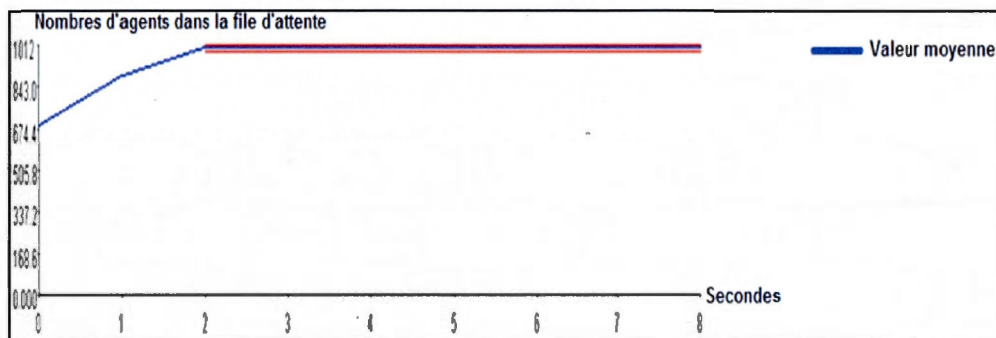


Figure 5.4 – Nombre d'agents sur la *File d'attente 2* dans le *Routeur 1* en fonction du temps

Le simulateur estime le nombre moyen d'agents mobiles sur la file d'attente (*Agents en état retour*) à 999,0549. Cette valeur est inférieure à 1000 (la capacité maximale). Avec une capacité égale à 1000, le nombre des pertes au niveau de cette file tend vers zéro. La valeur 1000 est donc une bonne valeur pour que cette file fonctionne parfaitement.

5.2.2 – Au niveau des compteurs intelligents

Nous avons choisi le compteur intelligent représenté par le foyer *F1*. Comme nous l'avons déjà relaté, on trouve une file d'attente sur un compteur intelligent. Le simulateur à ce niveau donne le graphe de la figure 5.5.

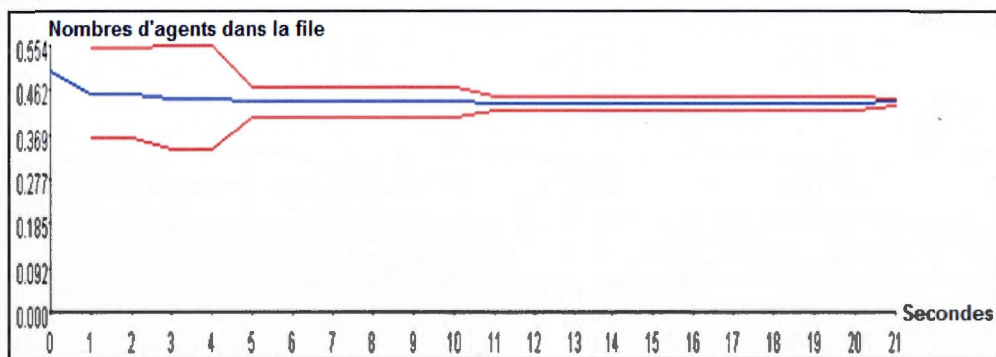


Figure 5.5 – Nombres d'agents sur la file d'attente dans un compteur intelligent en fonction du temps

Cette figure montre que le nombre moyen d'agents mobiles au niveau de cette file durant 21 secondes tend vers 0,4373. Ce nombre est inférieur à la taille proposée (qui est égale à 20). Puisque le nombre moyen d'agents dans la file d'attente est inférieur à 1, le nombre d'agents éliminés (qui correspond à la valeur de *drop* dans la simulation) à ce niveau est nul. On peut en déduire que la taille de cette file peut prendre une valeur faible (par exemple, 1 ou 2) afin de minimiser la taille de la mémoire réservée pour traiter un agent mobile.

CONCLUSION

Dans notre travail, nous avons investigué des tâches touchant à plusieurs aspects liés aux réseaux électriques intelligents.

Dans un premier temps, nous avons formulé les problématiques liées à ce type de réseau tout en présentant les différents travaux réalisés dans ce domaine et les besoins de ces réseaux en infrastructures de télécommunication et en services informatiques.

Dans le contexte de l'intégration des énergies renouvelables, nous avons étudié les architectures décentralisées de cette nouvelle génération de réseaux électriques ainsi que les aspects de l'offre et de la demande pour intégrer ces énergies nouvelles.

Le rôle important des compteurs intelligents a été mis en évidence afin de présenter les fonctions de distribution et de contrôle qu'ils permettent de réaliser. Nous avons également proposé une architecture globale de réseau électrique intelligent qui se base sur des compteurs intelligents, des routeurs et une infrastructure sans fil.

Nous avons ensuite présenté notre solution d'infrastructure de télécommunication pour supporter les échanges entre nœuds du réseau et les foyers des consommateurs. Cette architecture nous a facilité l'intégration d'une procédure d'ordonnancement pour gérer la distribution de l'énergie aux appareils dans les foyers des usagers du réseau. Nous avons notamment proposé un protocole d'échange et d'ordonnancement pour faire face aux problèmes de chute de tension.

Pour pallier le manque d'énergie et de l'intégration des énergies produites par les clients eux-mêmes, nous avons proposé un mécanisme d'échanges dans le cadre d'une procédure de négociation de l'offre et de la demande d'énergie. Cette solution se base sur les agents mobiles qui parcourent le réseau, à la recherche des fournisseurs qui satisfont les patterns de consommation des clients requérants. Ces agents mobiles sont pris en compte par un *middleware* installé dans les routeurs et les compteurs intelligents. Un protocole de routage

de ces agents a été mis en œuvre pour permettre de les diriger vers les foyers qui offrent de l'énergie et de les ramener à leurs foyers d'origine. Des services ont été implantés dans cette infrastructure pour permettre d'offrir ces fonctions de contrôle et de monitoring de tout le réseau.

L'utilisation des services Web dans les requêtes de consultation et de monitoring sur notre architecture a donné des performances que nous avons évaluées par simulation. Avec cette solution, en utilisant des appareils mobiles, les usagers du réseau peuvent consulter en temps réel et à distance la consommation énergétique des appareils dans leurs foyers. Ces services permettent également à des opérateurs du réseau électrique de consulter la consommation énergétique en temps réel d'une zone (groupement de foyers).

L'architecture à base d'agents mobiles et les structures proposées des compteurs intelligents et des routeurs nous ont permis de voir que l'intégration de la technologie des agents mobiles dans le domaine des réseaux électriques intelligents peut résoudre plusieurs problèmes actuels. Ils évitent d'accroître les interactions dans le réseau en même temps qu'ils permettent les prises de décision.

Dans ce travail, notre objectif n'était pas été de proposer une solution à tous les problèmes posés, car ce champ d'études pose de nombreux défis. Par exemple, nous n'avons pas considéré les aspects suivants qui, quoiqu'importants, pourraient faire l'objet de travaux futurs :

- *La conception et la réalisation des compteurs intelligents* – Cela nécessite de construire la circuiterie interne et des contrôleurs ; cet aspect est en dehors du cadre strictement informatique qui nous intéresse dans ce travail.
- *Les questions de sécurité (un élément critique dans ces types de réseaux)* – Il faut implanter des mécanismes rigoureux pour prévenir les intrusions dans le réseau et les foyers, et éviter ainsi la subtilisation de l'énergie ou simplement l'accès par des personnes non autorisées à des informations privées sur la consommation.

- *Une étude systématique des données de consommation stockées dans les bases de données de compteurs* – Ces données sont généralement de grande taille (de l'ordre des téraoctets par semaine par foyer). Des techniques de prospection de ces données (*data mining*) pourraient aider dans ce sens.
- *La conception de capteurs sur les appareils électroménagers afin de rendre possible la communication avec les compteurs intelligents.*
- *Des procédés de négociation de l'offre et de la demande* – Ce type de problèmes nécessite de recourir à des algorithmes complexes et des procédures d'optimisation. Dans notre travail, nous nous sommes contentés de comparer le pattern de consommation fourni par un requérant aux données offertes par des fournisseurs potentiels.

ANNEXES

1)

Algorithme de négociation:

- L'état de variables d'agent mobile dans le foyer initial avant d'être envoyé vers le premier routeur

Initial=true; // l'agent est dans le foyer initial (S'exécute dans le foyer initial)

Retour=false; // l'agent n'est pas dans état de retour vers le foyer initial (S'exécute dans le foyer initial)

Envoyer(Agent); // Vers le routeur associé à la zone où se trouve le foyer initial (S'exécute dans le foyer initial)

- Le code qui s'exécute sur chaque terminale visité (Routeur ou foyer)

Début // s'exécute sur chaque terminale visité (Routeur ou foyer)

Si (m="R"); // l'agent est dans un routeur

Initial=false; // l'agent à quitter le foyer initial

Si (Retour=false)

Si (Existe de Fournisseurs dans la zone)

Dupliquer_Agent(); // duplication d'agent à deux copie Agent(A) et Agent(B)

Agent(A).Vecteur_Routage.Empile(Adresse_Routeur); //Empiler l'adresse du routeur courant

Agent(B).Vecteur_Routage.Empile(Adresses_Fournisseurs); //empiler les adresses de foyers qui offre le service de vente dans la zone

Envoyer (Agent(A)); // Vers les routeurs voisins

Envoyer (Agent(B)) ; // Vers l'entête du vecteur de routage

Sinon // Pas de fournisseurs d'énergie dans la zone

Agent.Vecteur_Routage.Emile(Adresse_Routeur);

Envoyer(Agent); // Vers les routeurs voisins

Fin si

Sinon // Retour=true

Agent.Vecteur_Routage.Dépiler();

Envoyer(Agent); // Vers l'entête du vecteur de routage

Fin si

Sinon // l'agent est dans un foyer qui fournit l'énergie (m= 'F')

Agent.Vecteur_Routage.Dépiler();

Retour=True;

Si (initial=false)

Si (ServiceAccepté); // après la négociation, le service est accepté par l'agent

Empiler(Data_Service) ; // empiler dans le corps d'agent, les données qui représentent le service publié

Envoyer(Agent); //Vers l'entête du vecteur de routage.

Sinon

Envoyer(Agent); // Vers l'entête du vecteur de routage.

Fin si

Sinon

Agent.Vecteur_Routage.Dépiler();

Dépiler(Data_Service) ; // dépiler les données apportées par l'agent

Afficher (Data); //Affichées les données apportées.

Fin si

Fin si

Fin

2)

Code source (protocole de négociation en c#).

Ce code source représente un dispositif (routeur/compteur). Si la variable M=1 ce code représente un routeur, et si M=0 représente un compteur intelligent

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Xml;
using System.Text;

using System.Net.Sockets;
using System.Diagnostics;
using System.IO;
using System.Reflection;
using System.Net;
using Microsoft.CSharp;
using System.CodeDom.Compiler;
using System.Windows.Forms;
using System.Threading;
using System.Threading.Tasks;
namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        string Code;
```

```

public static string data = null;
public Form1()
{
    int i;
    InitializeComponent();
}

private void button1_Click(object sender, EventArgs e)
{
    ServerStart();
}

//-----middleware
public void MidellWare(string AgentCode, string CodeExec, string Vect,string Data,string
Retour,string Signature,string Initial)
{

    string client = @" private void SendMessage(string Ip,string code)
    {

        byte[] data = new byte[1024];
        string input, stringData;
        IPEndPoint ipep = new IPEndPoint(
            IPAddress.Parse(Ip), 9050);

        Socket server = new Socket(AddressFamily.InterNetwork,
            SocketType.Stream, ProtocolType.Tcp);

        try
        {
            server.Connect(ipep);
        }
        catch (SocketException t)
        {
            Console.WriteLine("Unable to connect to server.");
            Console.WriteLine(t.ToString());
            return;
        }

        int recv = server.Receive(data);
        stringData = Encoding.ASCII.GetString(data, 0, recv);
        Console.WriteLine(stringData);

        input=code;
        server.Send(Encoding.ASCII.GetBytes(input));
        data = new byte[1024];
        recv = server.Receive(data);
        stringData = Encoding.ASCII.GetString(data, 0, recv);
    }
}

```

```

        Console.WriteLine(stringData);
        Console.WriteLine("Disconnecting from server...");
        server.Shutdown(SocketShutdown.Both);
        server.Close();

    }

    ",
    //-----
    string Server = @"void StartListening()
    {

        int rcv;
        byte[] data = new byte[1024];
        IPEndPoint ipep = new IPEndPoint(IPAddress.Any, 9050);

        Socket newsock = new Socket(AddressFamily.InterNetwork, SocketType.Stream,
        ProtocolType.Tcp);

        newsock.Bind(ipep);
        newsock.Listen(10);
        Console.WriteLine("Waiting for a client...");
        Socket client = newsock.Accept();
        IPEndPoint clientep =(IPEndPoint)client.RemoteEndPoint;
        Console.WriteLine("Connected with {0} at port {1}",
            clientep.Address, clientep.Port);

        string welcome = "Welcome to my test server";
        data = Encoding.ASCII.GetBytes(welcome);
        client.Send(data, data.Length,
            SocketFlags.None);
        while (true)
        {

            data = new byte[1024];
            rcv = client.Receive(data);

            //execagent=Encoding.ASCII.GetString(data, 0, rcv);
            if (rcv == 0)
                break;

            Console.WriteLine(
                Encoding.ASCII.GetString(data, 0, rcv));
            client.Send(data, rcv, SocketFlags.None);
        }
        Console.WriteLine("Disconnected from {0}",
            clientep.Address);
        client.Close();
        newsock.Close();
    }
    //-----resend

```

```
ClientStart();
StartListening();
```

```
}";
```

```
        List<string> Agent = new List<string>();
        string codeexec = AgentCode;
        Agent.Add(@"
using System.IO;
using System.Text;
using System.Collections.Generic;
using System.Globalization;
using System.Security.Permissions;
using System.Security.Principal;
using System;
using Microsoft.CSharp;
using System.Net.Sockets;
using System.Windows.Forms;
using System.Linq;
using System.ComponentModel;
using System.Net;
using System.Threading;
using System.Threading.Tasks;
using System.CodeDom.Compiler;
using System.Diagnostics;
```

```
namespace Foo
{
    public class Bar
    {

        public void Analiser()
        {
```

```
//----- DECLARATION DU ROUTEUR
List<string> IpRouteurs=new List<string>();
List<string> IpFoyesConnectes=new List<string>();
string Vect;
string Vect2="";
string VectT;
string CodeS;
string Data;
string MessageVersFoyer;
string MessageProdcast;
int M=1; //un routeur M=1
string Retour;
string No="true";
string Signature;
```



```

string Initial;
string IpaRenvoi;
string[] T;
string IpFoyerNext="";
string IpLocal="IpLocal#";
//-----Les Ip de routeurs voisins si le cas d'un routeur
    IpRouteurs.Add("192.168.2.11"); // ici mettre une ip d'un routeur voisin
//IpRouteurs.Add("IpR2"); // une autre ip d'un routeur voisin
//-----
IpFoyesConnectes.Add("IpF1"); //--ip de foyer1
IpFoyesConnectes.Add("IpF2"); //---ip de foyer2
//-----
Retour=" + Retour+@";
Signature="+Signature+@";
Initial="+Initial+@";
Vect=" + Vect+ @";
VectT=" + Vect + @";
Data=" + Data+@";
" + AgentCode+CodeExec + @";

MessageVersFoyer=CodeS+"@"+"\""+Vect+"\""+@"+"\""+Data+"\""+@"+"\"
"+"\""+Retour+"\""+@"+"\""+Signature+"\""+@"+"\""+Initial+"\"";
MessageProdcast=CodeS+"@"+"\""+Vect2+"\""+@"+"\""+Data+"\""+@"+"\"
"+"\""+Retour+"\""+@"+"\""+Signature+"\""+@"+"\""+Initial+"\"";

if(M==0)
{
    T = Vect.Split('#');
    if(T.Length>1)
    {
        Vect="";
        for(int i=0;i<T.Length-2;i++)
        {
            Vect=Vect+T[i]+"#";
        }
    }

    MessageVersFoyer=CodeS+"@"+"\""+Vect+"\""+@"+"\""+Data+"\""+@"+"\"
    "+"\""+Retour+"\""+@"+"\""+Signature+"\""+@"+"\""+Initial+"\"";
    IpFoyerNext=T[T.Length-3];
}

//-----Si un routeur envoi aux autres retours et foyers
if(M==1) //Cas routeur
{
    if(Retour=="false"){
        if(IpRouteurs.Count>0)
        IpRouteurs.ForEach(delegate(String Ipr)
        {
            //-----Envoi au autres routeur
            SendMessage(Ipr,MessageProdcast);

```

```

MessageBox.Show("Message envoyer a:" + Ipr);

});

//----envoi au foyer connecté avec lui (Le dernier dans sa liste exp: si il a ip1,ip2,ip2 il envoi à ip3)

if(IpFoyesConnectes.Count>0)
//SendMessage(IpFoyesConnectes[IpFoyesConnectes.Count-1],MessageVersFoyer);
MessageBox.Show(IpFoyesConnectes[IpFoyesConnectes.Count-1]);

}
if(Retour=="true")
{
T = Vect.Split('#');
if(T.Length>1)
{
Vect="";
for(int i=0;i<T.Length-2;i++)
{
Vect=Vect+T[i]+"#";
}
}

MessageVersFoyer=CodeS+" @ "+"\""+Vect+"\""+"@ "+"\""+Data+"\""+"@ "+"\""+
Retour+"\""+"@ "+"\""+Signature+"\""+"@ "+"\""+Initial+"\"";
IpFoyerNext=T[T.Length-3];
//SendMessage(IpFoyerNext,MessageVersFoyer);
}

}

//-----Si un foyer ajoute data et envoyer aux eutres foyes ou au routeur
if(M==0) //Cas foyer
{

//SendMessage(IpFoyerNext,MessageVersFoyer);

}

}

//-----
string GetIp(List<string> ListIp) // lire les ip des foyers qui offre le servive de vente
{
string Sortie="";
ListIp.ForEach(delegate(String Ip)
{
//-----Envoi aux autres foyers connectés
Sortie=Sortie+Ip+"#";

});

```

```

return Sortie;
}
string GetIpRV()// Lire ip des routeurs voisin
{

return ""IpRV#"";
}
string GetDtata()// Lire ip des routeurs voisin
{

return ""Data2#"";
}
" + client + @"
    }
}
    );

```

```
//-----
```

```

Dictionary<string, string> providerOptions = new Dictionary<string, string>
{
    {"CompilerVersion", "v4.0"}
};
CSharpCodeProvider provider = new CSharpCodeProvider(providerOptions);

CompilerParameters compilerParams = new CompilerParameters
{

    // *** Start by adding any referenced assemblies

    GenerateInMemory = true,
    GenerateExecutable = false
};
compilerParams.ReferencedAssemblies.Add("System.dll");
compilerParams.ReferencedAssemblies.Add("System.Windows.Forms.dll");
compilerParams.ReferencedAssemblies.Add("System.Core.dll");

```

```

CompilerResults results = provider.CompileAssemblyFromSource(compilerParams,
Agent[0]);

```

```

object o = results.CompiledAssembly.CreateInstance("Foo.Bar");
MethodInfo mi = o.GetType().GetMethod("Analiser");

```

```

        mi.Invoke(o, null);

    }

//-----Start Compteur/Routeur
private void ServerStart()
{
    int recv;
    byte[] data = new byte[1024];
    IPEndPoint ipep = new IPEndPoint(IPAddress.Any,
        9050);

    Socket newsock = new
        Socket(AddressFamily.InterNetwork,
            SocketType.Stream, ProtocolType.Tcp);

    newsock.Bind(ipep);
    newsock.Listen(10);

    Socket client = newsock.Accept();
    IPEndPoint clientep =
        (IPEndPoint)client.RemoteEndPoint;

    string welcome = "Started";
    data = Encoding.ASCII.GetBytes(welcome);
    client.Send(data, data.Length,
        SocketFlags.None);
    while (true)
    {
        data = new byte[1024];
        recv = client.Receive(data);

        Code = Encoding.ASCII.GetString(data, 0, recv);

        string[] T = Code.Split('@');

        if (T.Length > 1)
        {
            string CodeExec = T[0];
            string Vect = T[1];
            string Data = T[2];
            string Retour = T[3];
            string Signature = T[4];
            string Initial = T[5];
            string CodeSource = "CodeS= \"" + CodeExec + " \"";

```

```

        MidellWare(CodeSource, CodeExec, Vect,Data,Retour,Signature,Initial);
    }
    if (recv == 0)
        break;

}

client.Close();
newsock.Close();

ServerStart();
}

private void button2_Click(object sender, EventArgs e)
{
    ServerStart();
}

}

}

```

3)

Ce code représente la fonction qui démarre un appareil intelligent, et calcule en temps réel la charge consommée par un appareil dans un intervalle de temps. Cette fonction est utilisée par le service Web.

```

StartAppliance (int IdAppl,int Time)
if (Drop[Time] == 0) // CASE NO DROP
{
    Charge[IdAppl]=PowerON[IdAppl]*Appliance[IdAppl,Time]+PowerOFF[IdAppl]; //Consomation
    in one stap in real time
    ChargeGl[IdAppl]=ChargeGl[IdAppl]+Charge[IdAppl]; //Consomation Global For one Appliance
}
else // CASE DROP
{
    Charge[IdAppl]=PowerON[IdAppl]*Appliance[IdAppl,Time]*Priority[IdAppl,Time]/10 +
    PowerOFF[IdAppl];
}

```

```
//Consumation in one step in real time
ChargeGl[IdAppl]=ChargeGl[IdAppl]+Charge[IdAppl]; //Consommation Global For one
Appliance
}
```

Une autre fonction qui permet d'envoyer un ordre de commande à un appareil à travers le port COM. Dans exemple nous envoyons une chaine « Start » sur le port COM, et on affiche un texte qui indique que le message a été envoyé. Cette fonction est représentée par le code suivant.

```
void StartAppl(string ApplCode)
{
    //Création d'une communication sur un port com
    SerialPort sp = new SerialPort();
    sp.Open();
    sp.WriteLine("start");
    sp.Close();
    Console.WriteLine("Message Envoyé vers:" +Convert.ToString(ApplCode));
}
```


BIBLIOGRAPHIE

- [1] C. Bissieres, « Distribution électrique et sécurité, » [En ligne]
<http://cbissprof.free.fr/telechargements/tsiris/cours/distributionelectrique.pdf>. [consulté le 26 avril 2012].
- [2] F. T, « Securing SCADA Infrastructure », Forinet, no IWP-SCADA-R1-201010, p. 2, 10, 2010.
- [3] W. Wies, « Design of an Energy Standalone Distributed Generation system Employing Renewable Energy Sources and Smart Grid Technology as a Student Design Project », Power and Energy Society General Meeting, 2010 IEEE, p. 1-8, july 2010.
- [4] Caramanis, « Load Participation in Electricity Markets : Day-ahead and Hour-Ahead Market Coupling with Wholesale/Transmission and Retail/Distribution Cost and Congestion Modeling », First IEEE International Conference on Smart Grid Communications (SmartGridComm), p. 513 -518, oct. 2010.
- [5] M. Erol-Kantarci, H. T. Mouftah, « Pervasive Energy Management for the Smart Grid: Towards a Low Carbon Economy, » Pervasive Communications Handbook, Eds. S. I. A. Shah, M. Ilyas, H. T. Mouftah, CRC Press, 2011.
- [6] A. D. a. V. Balakrishnan, « Energy Service Companies 40; Escos 41; To Optimize Power In Peak Demand Period In Hybrid Energy System: An Impact On Climate Change » Green Technologies Conference, 2010 IEEE, p. 1-10, april 2010.
- [7] Vale, « Distributed Energy Resources Management with Cyber-Physical Scada in the Context of Future Smart Grids », 15th IEEE Mediterranean Electrotechnical Conference MELECON, p. 431-436, April 2010.
- [8] A. K. a. A. Annaswamy, « Decision and Control (CDC) », 2010 49th IEEE Congress Centre Firenze, Italy Conference #20427, p. 194 -199, Dec. 2010.

- [9] Yang-Xin, « Research on Smart Appliances Control Protocol », International Workshop on Education Technology and Computer Science (ETCS), vol. 2, p. 551-554, March 2010.
- [10] J.-H. Shang-Wen Luan, « Development of a Smart Power Meter for Ami Based on Zigbee Communication », International Conference on Power Electronics and Drive Systems, 2009, PEDS 2009, p. 661-665, Nov. 2009.
- [11] J. H. a. W.-H. K. Es6, « Technologies for Smart Grid and Smart Meter », IEEE International Conference on Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011, p. 533, Feb. 2011.
- [12] Hochgraf, « Smart Grid Charger for Electric Vehicles Using Existing Cellular Networks and SMS Text Messages », Smart Grid Communications (SmartGridComm), 2010 First IEEE Int. Conference, p. 167-172, oct. 2010.
- [13] J. E. Samarakoon, « Smart Metering and Self-Healing of Distribution Networks », IEEE International Conference on Sustainable Energy Technologies (ICSET), pp. 1-2, dec. 2010.
- [14] S. G. a. P. L. Qiqi Zhang, « Smart City Grid :The start to develop smart grid », E-Product E-Service and E-Product E-Service and E-Entertainment (ICEEE), International Conference, pp. 1-4, nov. 2010.
- [15] Zhao Feng, « Study on the Application of Advanced Broadband Wireless Mobile Communication », In Power System Technology (POWERCON), International Conference, pp. 1-6, oct. 2010.
- [16] C. E. Kalogridis, « Privacy for Smart Meters : Towards undetectable appliance load signatures » Smart Grid Communications (SmartGrid-Comm), 2010 First IEEE International Conference, p. 232-237, oct. 2010.
- [17] Y. Shohoud, « Introduction to WSDL », 2001. [En ligne]. Available : <http://www.ijeei.org/docs-19158807814f8582920653e.pdf>. [consulté le 17 avril 2012].

- [18] U. Ali, S. Nawaz et N. Jawad, « A Real-Time Control System for Home/Office Appliances Automation, From Mobile Device Through GPRS Network », Electronics, Circuits and Systems, 2006. ICECS '06, 13th IEEE International Conference, n° 110.1109/ICECS.2006.379923, p. 854-857, 10-13 Dec. 2006.
- [19] « Consommations_appareils_electromenager_FA », Le centre urbain Asbl – De Stadswinkel vzw, 16 06 2011. [En ligne]. Available :

http://www.curbain.be/fr/energie/information/consommations_appareils_electromenager_FAC.php. [consulté le 13 avril 2012].
- [20] P. G. Serazzi, « Java Modelling Tools » Dipartimento di Elettronica e Informazione Politecnico di Milano – Italy, [En ligne]. Available: <http://jmt.sourceforge.net/>. [consulté le 22 juin 2012].
- [21] L. A. a. Z. Lukszo, « Smart Metering for the Future Energy Systems in the Netherlands », Critical Infrastructures, 2009. CRIS 2009. Fourth International Conference, pp. 1-7, 27. 30/04/2009.
- [22] J. F. a. E. G. M.C. Caramanis, « Load Participation in Electricity Markets : Day-ahead and Hour-Ahead Market Coupling with Wholesale/Transmission and Retail/ Distribution Cost and Congestion Modeling », Smart Grid Communications (SmartGridComm), 2010 First IEEE Int. Conf. , p. 513-518, oct. 2010.
- [23] W. C. a. W. Tseng, « An Agent Model for the Control of Smart Appliances », In Fuzzy Information Processing Society, 2007, NAFIPS '07, Annual Meeting of the North American, p. 95-99, june 2007.
- [24] C. Abbey, « The Hydro-Québec Distribution Smart Grid Zone », Hydro-Québec Research Institute (IREQ), 2011.
- [25] Y. T. Saifur Rahman, « Intelligent Distributed Autonomous Power Systems (IDAPS) », IEEE PES Summer Meeting in Tampa, Florida, June 24-28, 2007.
- [26] M. A. H. H. Hiromu Kobayashi, « Integrated Control Techniques of Supply and Demand for Japanese Style Smart Grid », n° 110.1109/PES.2010.5590148, p. 1-7, 25-

29 July 2010.

- [27] T. Perumal, A. Ramli et C. Leong, « Design and Implementation of SOAP-Based Residential Management for Smart Home Systems » *Consumer Electronics, IEEE Transactions*, vol. 54, n° 110.1109/TCE.2008.4560114, p. 453-459, May 2008.
- [28] C.-M. Huang, H.-H. Ku et C.-W. Lin, « Design and Implementation of a Web2.0-Based Home-Appliances Control Service Platform », *Pervasive Systems, Algorithms, and Networks (ISpan)*, n° 110.1109/I-SPAN.2009.72, p. 110-115, 14-16 Dec. 2009.
- [29] EnergyTracking, « Web Enabled Meter », Energy Tracking, 2007. [En ligne]. Available : <http://www.energytracking.com/wem.htm>. [Accès le 12 06 2012].
- [30] At-A-Glance, « Cisco Smart Grid (Solutions for the Next-Generation Energy Network) » Cisco Systems, 2009.
- [31] R. N. N. E. M., « Oracle Solutions for Smart Grid » Oracle Utilities, p. 1-24, Sept. 2, 2009.
- [32] J. Wu, « China Smart Grid Development Model and Industry Prospect », China Smart Engineering Association, p. 1-43, August 10, 2011.
- [33] A. Aggarwal, S. Kunta et P. Verma, « A Proposed Communications Infrastructure for the Smart Grid », *Innovative Smart Grid Technologies (ISGT)*, 2010, n° 1 10.1109/ISGT.2010.5434764, p. 1-5, 19-21 Jan. 2010.
- [34] J. CARLO, « The IEEE 802.16 Working Group on Broadband Wireless Access Standards » 12 11 2011. [En ligne]. Available: <http://www.ieee802.org/16/>. [consulté le 08 mars 2012].
- [35] C.-L. Fok, G.-C. Roman et C. Lu, « Mobile Agent Middleware for Sensor Networks: an Application Case Study », *Information Processing in Sensor Networks*, 2005. IPSN 2005. Fourth International Symposium on, n° 110.1109/IPSN.2005.1440953, p. 382-387, 15 April 2005.

- [36] P. D. a. S. Valaee, « 802.16 Mesh Networks », December 8, 2006.
- [37] R. McCaughern, « Consultation Paper on Broadband over Power Line (BPL) Communication Systems », Spectrum Engineering Branch, n° 1SMSE-005-05, p. 1-12, July 2005.
- [38] S. W. Loke, « Mobile Agent Technology for Enterprise Distributed Applications: An Overview and an Architectural Perspective », CRC for Distributed Systems Technology Monash University, Caulfield Campus, PO Box 197, Caulfield East, Victoria 3145, Australia, p. de 1 à 28.
- [39] P. S. Ningning Hu, « Estimating Available Bandwidth Using Packet Pair Probing », School of Computer Science Carnegie Mellon University Pittsburgh, PA 15213, n°1CMU-CS-02-166, September 9, 2002.
- [40] S. J. a. R. W. K. T. N. Rob van Gerwen, « Smart Metering », The Global Community for Sustainable Energy Professionals, p. 1 à 9, july 2006.
- [41] C. O. D. F. Jianwen Shao, « Smart Home Appliance Control », STMicroelectronics, Inc. Schaumburg, IL USA, p. pp.1-6, 5-9, Oct. 2008.
- [42] «Tarif-residentiel » Hydro-Québec, 2012. [En ligne]. Available : <http://www.hydroquebec.com/residentiel/tarif-residentiel.html>. [consulté le 13 avril 2012].
- [43] « Conseils_eclairage », ADEME, [En ligne]. Available : http://www.ademe.fr/bretagne/actions_phares/energie_maitrise/conseils_eclairage.asp. [consulté le 13 avril 2012].
- [44] Michael Rathmair, Jan Haase" Simulator for Smart Load Management in Home Appliances" SIMUL 2012 : The Fourth International Conference on Advances in System Simulation, Vienna University of Technology, Institute of Computer Technology 1040 Vienna, Gusshausstrasse 27-29/384, Austria

- [45] G.W.Hart, "Nonintrusive Appliance Load Monitoring," SICE Annual Conference 2008. Proceedings of the IEEE 1992, pp.1870--1891.
- [46] S.S. Kuruppu, Student Me « Smart Meter Based Non-Intrusive Appliance Detection Algorithm for Local Real Time Feedback Purposes» IEEE, and N.Athula. Kulatunga, Member IEEE, IEEE PES ISGT ASIA 2012 1569525817.
- [47] Omid Roozmand, Mohammad Ali Nematbakhsh, Ahmad Baraani: An Electronic Marketplace Based on Reputation and Learning, Journal of Theoretical and Applied Electronic Commerce Research ISSN 0718–1876 Electronic Version VOL 2 / ISSUE 1 / APRIL 2007 / 1 - 17
- [48] <http://www.ia.hiof.no/softengin/ias/literature/ag/wwca.pdf> consulté le 06/03/2013
- [49] Lin, Fang-Chang : Systems, Man and Cybernetics, 2002 IEEE International Conference on: Cooperative Multi-Agent Negotiation for Electronic Commerce Based on Mobile Agents , 10.1109/ICSMC.2002.
- [50] Fac. of Comput. & Inf. Sci., Hosei Univ., Tokyo, Japan : A mobile negotiation agent embedded hybrid online purchasing system- P -227, Distributed Computing Systems Workshops, 2004. Proceedings. 24th International Conference, 10.1109/ICDCSW.2004.1284035
- [51] Dasgupta, Prithviraj ; Narasimhan, Nitya ; Moser, Louise E. ; Melliar-Smith, Peter M: MAgNET: mobile agents for networked electronic trading Knowledge and Data Engineering, IEEE Transactions on Volume: 11, Issue: 4 Digital Object Identifier: 10.1109/69.790796 Publication Year: 1999 , Page(s): 509-525